MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

AFIT/GST/OS/84M-2

AD-A141 217

An Improved Missile Combat Crew
Scheduling System Using the Simulation
Language for Alternative Modeling (SLAM)

Thesis

Capt D. H. Berg   Capt R. G. Nuss

AFIT/GST/OS/84M-2

MAY 1 7 1984

A

Approved for public release; distribution unlimited

84 05 15 050

AN IMPROVED MISSILE COMBAT CREW

SCHEDULING SYSTEM USING THE SIMULATION

LANGUAGE FOR ALTERNATIVE MODELING (SLAM)

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

by

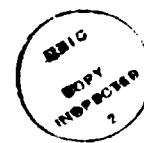David H. Berg, B.S.   Rodney G. Nuss, B.S.

Captain   USAF   Captain   USAF

Graduate Strategic and Tactical Sciences

March 1984

Preface

The purpose of the study was to design a missile combat crew scheduling program for use by the scheduler at strategic missile wings. Little previous research has been done in this area.

This research is limited in scope to the model design, program construction, validation, verification, and a preliminary analysis of the schedules produced by the program. The program is useful, as written, to the the wing scheduler for constructing initial monthly operations schedules. The program represents one part of a Decision Oriented Management System for missile wings.

We would like to thank our advisors, Lt Col Thomas D. Clark and Lt Col Palmer W. Smith of the Air Force Institute of Technology who have given this effort timely guidance and support. Deep gratitude is expressed to Maj James R. Coakley of the Air Force Institute of Technology for his aid in solving several computer problems and for listening. Finally, we wish to thank our wives, Christine Berg and Betty Nuss, for their support and patience through this effort.

Contents

List of Figures

List of Tables

List of Abbreviations/Acronyms

ACP   Alternate Command Post

AFB   Air Force Base

CPM   Critical Path Method

DNIA   Duty Not Including Alert

DOSS   Decision Oriented Scheduling System

DOT   Operations Training Division

DOV   Standardization and Evaluation Division

EA   Eligible for Alert

ERCS   Emergency Rocket Communications System

EWO   Emergency War Order

FIFO   First-In-First-Out

HVF   High Value First

ILP   Integer Linear Programming

LCC   Launch Control Center

LIFO   Last-In-First-Out

LP   Linear Programming

LVF   Low Value First

MCC   Missile Combat Crew

MCCM   Missile Combat Crew Member

MCDT   Multi-Criteria Decision Theory

MOE   Measure of Effectiveness

MPT   Missile Procedures Trainer

PERT   Program Evaluation and Review Technique

PLCC   Primary Launch Control Center

Q-GERT   Queueing Systems Graphical Evaluation and
         Review Technique

RSCM   Reserve Crew Member

RSM   Response Surface Methodology

SAC   Strategic Air Command

SCP   Squadron Command Post

SLAM   Simulation Language for Alternative Modeling

SMW   Strategic Missile Wing

Abstract

A missile combat crew scheduling program was developed using the SLAM simulation language. The program schedules missile combat crew members for alerts, training, and leave. The program was designed to build monthly schedules for a three-squadron strategic missile wing.

A performance measure was developed to evaluate alternative schedules that were developed by the program. The measure was developed by using Multi-Criteria Decision Theory techniques and alternatives were compared using response surface methodology. The performance measure is general in nature and can be modified to apply to any decision maker.

Analysis was conducted to determine the best internal factor settings for the program for a given performance measure. These internal factor settings are used to determine the next crew selected for alert.

Sensitivity analyses were conducted that include evaluating two additional performance measures, adding an assumed leave distribution, and changing the structure of the crew force.

The program develops feasible monthly schedules that meet the performance criteria. The internal factors were found to be robust across all three performance measures examined.

CHAPTER I

INTRODUCTION

BACKGROUND

The scheduling for missile operations crews at strategic
missile wings (SMW) is done manually with grease pencils and
wall boards. It is a time consuming process that requires
the full time efforts of several schedulers. The process is
inefficient for several reasons. The manual manipulation of
scheduling data causes errors, trends and statistics for
scheduling are difficult to analyze manually, and the sched-
uler has a difficult time seeing the "big picture." Because
of the information overload of the manual system, opportuni-
ties for improving the schedule are lost.

There are no measures of effectiveness (MOEs) that tell
the schedulers how good they are doing their jobs. There
are also no criteria for dealing with conflicting goals that
may force a scheduler to choose between one scheduling al-
ternative and another. Even the goals set forth for the
wing schedule are vague and non-quantifiable. Without a
clear set of goals, a quantifiable measure of effectiveness,
and a method of choosing between scheduling alternatives,
there is no incentive to produce a "good" missile operations
schedule. In fact, a "good" schedule has never really been
defined, and that is part of the problem.

The computerization of the missile scheduling process could reduce these problems in many ways. A computerized system that is designed to aid, not replace, the scheduler could save many man-hours in producing a high quality schedule. The computer could serve as a data base for scheduling information. After the scheduler has provided inputs concerning the status of the operational force and the constraints on the schedule for a given month, the computer can do the manual work of filling out the schedule and checking for conflicting entries. If a conflict exists and the scheduler wishes, the computer program can either solve the conflict according to some pre-established rule or allow the scheduler to solve the conflict manually.

Besides computerizing the scheduling process, several other areas must be addressed. Logical measures of effectiveness (MOEs) must be devised. These MOEs must be quantifiable so that the scheduler can tell if one particular alternative is better than another. Before these measures can be obtained, specific scheduling goals must first be developed. This type of information is obtained by using the techniques of Multi-Criteria Decision Theory (MCDT). With MCDT, the goals and MOEs for each missile wing can be developed in accordance with the wishes of the wing commander. Each wing may have different goals and MOEs, and each wing scheduler would incorporate these goals and MOEs to improve their particular schedule.

The computerized schedule could also improve the

scheduling of missile combat crew members (MCCMs) by in-
suring that the schedule was built as equitably as possible,
taking into account crew member's desires to the maximum
extent possible.

The ability of a missile crew scheduler to build a high
quality schedule with a grease pencil and a wall board is
limited. The two major problems that the scheduler en-
counters when he starts to build the schedule are:

a. The ambiguity and conflicting nature of the sched-
   uling objectives.

b. The lack of selection criteria for choosing among
   scheduling alternatives (Ref 2).

Berman points out that, "Even if the computational power
were available, schedulers still lack a measurement scheme
that provides all goals in a structured form and aids in
understanding the total level of wing performance. They
also need a method of indicating the effect of tradeoffs
among conflicting objectives on the performance measures.
Without such tools and techniques, the ability to find even
near-optimal solutions is lacking." (Ref 7:83)

Currently, schedulers can create one feasible schedule
for a missile operations wing per month. The scheduler has
no time to create several alternative schedules and select
the best schedule. Additionally, the scheduler does not
possess a means of comparing alternative schedules to de-
termine the best alternative. A computerized scheduling
system would allow the scheduler to create alternative

schedules quickly. If some criteria for judging alter-

natives are developed, the alternatives could be examined

and the best schedule selected.

PROBLEM STATEMENT

An efficient method to allow the scheduler to create

alternative schedules and select the best alternative in a

timely manner does not exist.

RESEARCH QUESTIONS

The following questions are what this research intends

to answer:

a. Is it possible to develop a computer model to build

    missile combat crew (MCC) schedules?

b. Is it possible to compare alternative schedules in

    a quantitative manner?

OBJECTIVES

The objectives of this research effort are as follows:

a. To develop a computerized scheduling model for MCC

    scheduling, that allows the wing scheduler to input

    data and build a number of alternative schedules

    quickly and efficiently.

Berman suggests a good reason to achieve this objective,

"If we could mechanize the schedule building process, even

by imitating the rules used by schedulers, we could examine

I-4

many schedules, and thus depict the interrelations of var-
iables and parameters." (Ref 7)

      b. To use this computer model to:

          1) Analyze alternative schedules.

          2) Determine the feasibility of using the model
to forecast future manning requirements.

SCOPE

There are two major assumptions used in this research
effort. First, alerts, weapon system training, Emergency
War Order (EWO) training, Missile Procedures Trainer (MPT)
sessions, and leave were designed to be scheduled by the
model. These events were seen as capturing the essence of
the schedule. Other events such as evaluations, hand gun
training, and physicals could be added, but these events do
not significantly effect the selection of the best alter-
native schedule.

Second, missile combat crew members were scheduled as
crews, not on an individual basis. This assumption leads to
a simplification of the scheduling problem, but there are
difficulties with this approach. These difficulties are ex-
amined in Chapter IV of this report.

SYSTEM STRUCTURE

It is essential to look at the structure of a strategic
missile wing (SMW) in depth to understand the task that

faces the wing scheduler.

There are three types of missile combat crews (MCCs) at an SMW. First, there are the crews in the training and evaluation (DOT and DOV) divisions of the wing. These crews instruct and evaluate the rest of the missile crew force. Each missile flight consisting of three to six MCCs is supervised by a flight commander. It is the flight commander's job to oversee the operation of the flight. He takes care of details such as administrative paperwork and general supervision of the MCCs in his flight. The flight commander is also expected to complete a certain amount of "office duty" every month. The flight commander takes care of problems and administrative details that occur in the missile squadron offices. The last type of MCC is the "line crew". The line crew is the basic operations resource at every SMW.

These three types of MCCs also perform alert duties at three different alert rates. The DOT/DOV crews normally have two alerts per month. The flight commanders have four alerts per month while the line crews usually have no more than eight alerts per month. The eight alert per month level is a Strategic Air Command (SAC) guideline for the SMW. This guideline can be exceeded with the wing commander's permission. Obviously, a high quality schedule is going to have the greatest effect on the line crew. The morale of the line crew member is essential to improving or maintaining the effectiveness of the missile combat crew force at an SMW.

The MCCs perform alerts at launch control centers
(LCCs). At the 341st SMW, Malmstrom AFB, Montana, for
example, the LCCs are located anywhere from 20 to 125 miles
from the support base. The distance involved to and from
alerts can also be a significant factor used in developing
an operations schedule. There are two types of LCCs at
every SMW. The primary LCC (PLCC) can be manned by any
qualified MCC. The squadron command post (SCP)/alternate
command post (ACP) can be manned only by SCP/ACP qualified
crew members. The DOT/DOV crews, some line crews, and some
flight commanders are normally ACP/SCP qualified.

There are three or four squadrons at each SMW. Every
squadron has five LCCs with one LCC in each squadron serving
as the SCP. In addition, one SCP serves as the ACP for that
wing. This means that there are a total of 15 to 20 LCCs at
each wing, with two to three SCPs and one ACP. There are
approximately 90 MCCs at a three-squadron missile wing when
the wing is fully manned. The alert load for each of the
crew types depends on the total number of crews and the num-
ber of ACP/SCP qualified crews at each wing. These are the
major resources that a wing scheduler has to use when he
builds an operations schedule.

The following is a list of major activities that the
scheduler must assign to the resources:

    a. Alerts.

    b. MCC Training.

        1) Emergency War Order Training (EWO).

2) Codes Training.

3) Weapon System Recurring Training.

c. Missile Procedures Trainer (MPT) Session(s).

d. Leave.

When scheduling MCC classroom training, several constraints are usually kept in mind by the scheduler. First, the number of class days is kept to a minimum. This is done to allow instructors more time to perform other tasks that they are responsible for. At the same time, the class size should be kept to a manageable level to allow meaningful and effective training for the MCCs. These two goals present the scheduler with conflicting alternatives. How the scheduler resolves that conflict has an effect on the operations schedule. Normally, each MCC receives this training on a monthly basis.

MPT sessions are classes spent in a simulator practicing EWO and weapon system checklist actions to keep the MCCs current and familiar with the weapon system. The amount of MPT time each MCCM receives is set by the Deputy Commander for Operations. This number is a function of how much MPT time is available. The number of MPT sessions varies from wing to wing. If an MPT is broken or other contingencies occur then the number of MPT sessions per MCC changes. Most MCCs receive one or two sessions, each of four to six hours per month. SAC regulations require that each MCC receive at least one MPT session every six months. However, the number of MPT sessions given to each MCC can vary widely between

SMWs.

Leave is usually restricted to no more than 21 days at a time for an MCC. This is a function of training requirements and manning considerations at each SMW. Leave is normally taken as a crew (integral leave), but there are exceptions to this policy. The number of crews allowed on leave at a given time is also a function of the manning level of the SMW. These are the operations resources and activities that the scheduler must deal with every time a schedule is built.

There are numerous other scheduled activities that the scheduler must contend with on a daily basis. The activities and resources listed are the wing schedulers primary concern. If these items are taken care of the other activities are minor in comparison.

The missile operations scheduler takes the inputs listed in the preceding paragraphs as a starting point for building the monthly schedule. These inputs are used by the scheduler in some heuristic algorithm that creates a schedule.

Additionally, the scheduler has the following constraints, as a minimum, in constructing the schedule:

    a. All launch control centers (LCCs) must be manned every day.

    b. All training must be administered to the appropriate crews.

There are many feasible schedules that a scheduler can construct within these constraints. A balance between

training effectiveness, resource utilization, and morale
must be maintained to build a schedule that works and is
satisfactory to everyone.

One major problem is the daily schedule changes that
occur during a month for one reason or another. Any effort
at computerization will certainly have to take these sched-
ule changes into account.

PREVIOUS RESEARCH EFFORTS

The discussion of previous research efforts in the area
of scheduling will be presented in two parts. The first
part will deal with the types of scheduling problems. The
solutions attempted for these problems are also listed. The
second part will deal with other pertinent information re-
lated to missile combat crew scheduling in particular.

SCHEDULING PROBLEMS: TYPES AND TECHNIQUES

The various types of scheduling problems will be dis-
cussed first. Problem types include: job shop scheduling,
maintenance scheduling, network scheduling, and cyclical
scheduling.

Job shop scheduling involves jobs or items flowing
through a machine shop. The object is to schedule each job
for the proper type of machine at the proper time as effi-
ciently as possible. The jobs may have to be done in a
particular order, or they may be done in random order,

I-10

depending on the job shop. Each job may be required to visit each machine in the job shop a certain number of times, or it may only visit some subset of the machines in the shop. Additionally, there may be several types of jobs which require different subsets of machines or different types of flow patterns. Algorithms to solve job shop problems usually use Integer Linear Programming (ILP) or heuristic techniques. These algorithms tend to be created for specific problems rather than a general problem type. The literature in this area deals with finding algorithms for new, specific problems, expanding old algorithms, or exploring how different algorithms can be used to solve the same problem.

Hosios and Rousseau (Ref 18) present a heuristic algorithm that schedules personnel to complete a set of activities every scheduling period. The activities are given at various times and locations. The algorithm also attempts to minimize the number of personnel at every location-activity-time period. The algorithm assumes that there is no preference by the personnel about the order of completing the activities.

MCCs must also be scheduled for various activities during the month. The order of the activities is more important in the missile scheduling problem, however. One objective of the scheduling program is to spread MCC alerts throughout the month. Hosios and Rousseau do not consider this problem. All activities are also not completed by

I-11

every crew. For example, only SCP qualified crews can perform alerts at the SCPs.

Martel (Ref 20) looks at jobs that can pre-empt each other and that have specific start and stop times. The problem is treated as one of entities flowing through a network with the objective to get the most entities through the network on time.

Pre-emption is not a capability that is needed in scheduling MCCs. The scheduling program is not concerned with finishing a project in an optimal amount of time. The program is concerned with allocating scarce resources in an efficient fashion.

New (Ref 25) discusses various dispatching rules for the job shop and how they effect the flow of jobs. A dispatching rule is a method for ranking jobs waiting to begin some process.

The idea of dispatching rules is an important concept in MCC scheduling. Deciding on the next crew to perform alert is a problem in ranking the crews that are waiting to be selected for alert. New examines dispatching rules in terms of finding a rule that will have all jobs completed in the shortest amount of time. Once again, this is not the problem for operations scheduling. The problem is how to allocate the crews in the best possible manner to complete alert requirements.

Maintenance scheduling problems tend to cover a broad spectrum of maintenance activities. Some problems deal with

which repair crew to send out to a certain location at a certain time. Other problems deal with scheduling items through a maintenance facility in some optimal way (similar to a job shop problem). Still other problems attempt to deal with the manpower required to maintain a particular item during some part of the item's life cycle. Some type of ILP is usually used to solve these problems, and a wide variety of problems can fit a certain program structure. Several problems have been solved using the simulation language Q-GERT. These problems dealt with scheduling items through a maintenance depot facility in the most efficient way (Ref 23). This is the first area to explore the use of simulation as a scheduling technique. There has been a considerable amount of literature on maintenance scheduling problems. The majority of this literature deals with the maintenance of power generation equipment.

Yamayee (Ref 32) is a good literature review of research in this area. This report concerns the scheduling of main- tenance for utility company power generating equipment. The objective of the scheduling program is to minimize costs. To accomplish this objective, preventative maintenance is employed; repairing the equipment before it breaks. There are several uncertainties associated with a long term sched- uling program in this area. These uncertainties include:

    a. Expected power load.

    b. Fuel prices.

    c. Fuel supplies.

       d. Maintenance crew availability.

       e. Reliability of equipment (Mean time between

           failures).

Yamayee presents and discusses integer programming, dynamic
programming, and heuristics as approaches to solving this
scheduling problem.

The objective of maintenance scheduling is to minimize
cost by either reducing manpower requirements or reducing
repair costs. To do this requires an optimal allocation of
repair crews and money to the equipment being maintained.
The objective of MCC scheduling is to allocate crews to
alert and training activities in an optimal fashion. Both
maintenance and MCC scheduling deal with an optimal allo-
cation of resources to activities. Both types of scheduling
use similar methods in solving the problem for this reason.

Network scheduling involves finding the critical path
through a series of activities that must be completed in
some time dependent manner. This type of problem is nor-
mally concerned with construction or manufacturing processes
that must be performed in a sequence. The object is to
identify the most efficient method of completing the activ-
ities. Another objective is to identify where efforts
should be concentrated to improve the process and save time.
PERT and CPM are the two techniques most associated with
this type of activity. There has also been a considerable
amount of research done in this area. Most of this work was
done about 10 years ago.

Chiplunkar, Mehndiratta, and Khanna (Ref 9) develop a
design for optimizing the routes of refuse collection trucks
in a city. A heuristic algorithm is used to analyze the
network.

McGough (Ref 21) discusses the use of PERT/CPM to build
activity schedules for construction projects.

Crandall and Woolery (Ref 11) and Crandall (Ref 10) look
at stochastic network schedules. These types of schedules
have some probability distribution associated with project
completion times and the criticality of activities. This
leads to a set of probable critical paths. Monte-Carlo
simulation was used to generate the models. Crandall (Ref
10) compares Monte-Carlo simulation and PMET in generating
schedules with stochastic activities. PMET is another com-
puter algorithm used to analyze stochastic networks.

Cunto (Ref 12) describes a heuristic algorithm developed
to schedule boats for sampling oil wells. Each boat is
assigned a specific route and set of oil wells to visit.
The oil wells are sampled at specific times each period.
The algorithm has improved the efficiency of this sampling
effort.

Network scheduling is similar to job shop scheduling in
that it seeks to optimize the flow of goods or activities
through a process. This type of scheduling deals with im-
proving the time it takes to complete the network, not the
amount of resources it takes to complete a task. For this
reason network scheduling is not helpful in building an MCC

I-15

scheduling program. One idea from network scheduling is important, however. MCCs can be thought of as entities flowing through a network at a constant rate. These entities can be assigned to various events. This concept has applications to an MCC scheduling program as is shown in Chapter II.

Cyclical scheduling involves the scheduling of people for shift work or a schedule of on and off days, such as nursing schedules. This type of schedule is the closest to the type of problem that involves crew scheduling. Several good articles were written on this subject in the mid-1978's. Warner (Ref 31) developed a nursing schedule that took into account weekends off, work stretches (consecutive work days), single days off, and undesirable work patterns (back-to-back shifts).

In this article, Warner also addresses the need for a high quality schedule to keep morale high and to improve working conditions. He also includes a method of weighting criteria to determine the "goodness" of a schedule. This technique can also be used as a measure of optimality for crew schedules. This topic will be discussed in detail in Chapter IV. ILP and heuristics were the main techniques used to solve these types of problems. The majority of the literature deals with different ways to apply these techniques to slightly different variations of the basic problem.

Mageath (Ref 19) and Arthur and Ravindran (Ref 3) have

also written articles on the nurse scheduling problem.

Mageath proposed a simple heuristic approach after rejecting
a mathematical programming one. The mathematical program
that was conceived to handle the problem had over 8800 vari-
ables and 400 constraints. To carry out such an analysis
would have taken more computer resources than the hospital
wished to provide.

Arthur and Ravindran proposed a goal programming ap-
proach to the nurse scheduling problem. A goal program in-
corporates several goals into the objective function, seek-
ing to simultaneously optimize all the goals. This provides
an added degree of flexibility to the scheduling program in
that different goals can be substituted into the program if
desired. The goal program is used to determine work days,
and a heuristic (similar to Warner's and Mageath's) is used
to schedule the work shifts on each day.

Bartholdi (Ref 5) looks at staff scheduling in general.
He outlines an algorithm that will minimize the work force
required on every shift and give two consecutive days off
every week. This algorithm is limited in its usefulness to
general scheduling problems.

Warner's use of a set of weighted criteria is very sim-
ilar to goal programming. A set of weighted measures of
effectiveness is generally used with Multi-Criteria Decision
Theory (MCDT).

The work of these authors deals with one aspect of MCC
scheduling; the scheduling of alerts is a cyclical sched-

uling type problem. Both heuristics and mathematical pro-
gramming also apply to MCC scheduling, as they apply to
cyclical scheduling.

However, cyclical scheduling falls short of providing
all the answers for the MCC scheduling problem. First, more
than one type of activity must be scheduled for MCCs, not
just a single shift per day. The MCC schedule also includes
classroom training and MPT sessions. This can be thought of
as three different cyclical schedules operating simultane-
ously. Second, leave is not considered in the cyclical
scheduling literature. Bartholdi is the most ambitious au-
thor in this area by suggesting a means of scheduling two
consecutive days off for personnel. Leave for the crew
force is actually some type of probability distribution in-
volving the starting date, the duration, and the number of
crews taking leave at the same time. This problem exceeds
the ability of the cyclical scheduling algorithms.

The difficulty with using these problem types as a guide
to MCC scheduling lies in the techniques used, not the prob-
lems themselves.

Scheduling problems are normally solved by using Integer
Linear Programming (ILP), Dynamic Programming, or heuristics
(Ref 32). These techniques are applied to the different
types of scheduling problems.

ILP techniques are not suitable for crew scheduling for
the following reasons:

a. Number of constraints involved,

b. Number of variables involved,

c. Lack of flexibility in dealing with schedule changes, and changes to the input variables and,

d. Inability to deal with the stochasticity of the scheduling process.

Dynamic programming runs into similar problems dealing with the so called "Curse of Dimensionality" (Ref 17:285).

Heuristics offer one hope for the scheduling of missile crews. The scheduler currently builds the schedule by using a set of heuristics that he has developed. The use of simulation in building a missile schedule would take this set of heuristics and automate them. This would give the scheduler a method of comparing alternative schedules and would also allow analysis to be performed on the schedule.

Several additional research efforts were conducted that should be discussed. The first two reports deal with Strategic Air Command (SAC) air crew scheduling in a quantitative fashion (Ref 6,7). These reports, by Berman, propose the creation of a Decision Oriented Scheduling System (DOSS) to take care of scheduling shortcomings. The problems of aircrew scheduling in SAC are similar to missile crew scheduling problems.

The DOSS objective would be to create an initial schedule and identify its performance level. This initial schedule would then be changed to improve its performance to the desired level. According to Berman, the DOSS should be able to have flexibility, look ahead, and handle major and minor

unforeseen disruptions.

Berman suggests that, "While there are no known optimi-
zation techniques, sub-portions of the scheduling problem
may be amenable to known optimization algorithms, which
should be used when possible (Ref 6)." Berman also goes on
to say that Multi-Criteria Decision Theory (MCDT) may play a
part in the optimization process.

A third report, by Angell, was written specifically
about missile crew scheduling (Ref 2). This report spells
out a process, using MCDT, that could be used to optimize a
crew schedule. In addition, Angell discuss scheduling prob-
lems pertinent to MCC scheduling and a systematic approach
for coping with these problems. This report will be ref-
erenced in other portions of this research effort.

Fallon (Ref 15) and Hershauer and Ebert (Ref 16) deal
with an area of study called Decision Support Systems.
These systems are computerized models that use cybernetic
processes to mimic decision makers. The decision makers
attitudes and methods of making decisions (heuristics) are
input into a computer program that aids the decision maker.
The idea is to give the decision maker a tool that consist-
ently supports his own thought processes. This idea can be
applied to scheduling in particular. Fallon uses these
ideas to build a Decision Oriented Management System that
helps analyze scheduling decisions at SAC B-52 wings.

Decision Support Systems are designed to be management
analysis tools that incorporate scheduling programs. The

scheduling program and a data base that contains information on resources are combined to form a DOSS. This allows the scheduler to recall needed information from the data base, consider the data presented, make a decision, and input the decision into the schedule. The schedule is updated and can be judged by some criteria (a measure of the schedule's performance) to determine the effect of the decision on the quality of the schedule. In this way schedules can be built that reflect the decision maker's desires in an efficient manner.

In other words, the scheduling program proposed in this research effort is a part of a DOSS that could effect scheduling operations at the strategic missile wing.

Several different scheduling techniques are being applied to several different types of problems in scheduling today. Most of these techniques are, unfortunately, not what is needed to build a missile crew schedule. The works of Berman and Angell outline what is needed in the way of an effective crew scheduling algorithm. It is the aim of this research effort to carry through with their work to develop that algorithm.

METHODOLOGY

This research effort will use the system science paradigm as a general method of accomplishing the research goals. This paradigm is set forth in the work of Schoderbeck, Schoderbeck, and Kefalas (Ref 29). There are three

parts to the paradigm:

    1) Conceptualization,

    2) Analysis and measurement, and

    3) Computerization.

The following paragraphs will discuss each part in more detail.

Conceptualization involves looking at the problem as a system that interacts with its environment. The idea is to try to capture the complex interaction of the system and its environment as a series of models. These models get successively narrower and more defined. This can be thought of as traveling down a "cone of resolution" from the top (a broad model) to a point where the model is specific enough to provide the type of understanding desired (Ref 29:297). The broad model is conceptualized as a set of inputs that are processed to provide some output. The inputs are brought into the system from the environment where the system processes them. The processed inputs are then returned to the environment as outputs. The object of the model is not to describe every detail of the environment and the system, but to model the important aspects of both. Once this broad model is built, the focus of the model is narrowed as the researcher tries to define the analysis and measurement of the critical areas of the model.

Analysis and Measurement tries to quantify the changes in the state of the elements of the model. The object is to gain an understanding of the underlying elements of the

system. Once the researcher understands the process, a

model is built that describes it. Once again, the model

does not have to exactly duplicate the system, but it must

duplicate the system to the degree that it achieves the

desired objectives of the research. In this part of the

paradigm, there are several things that the researcher must

decide:

1) The language in which the results of the model will
   be expressed (LANGUAGE).

2) Under what conditions and in what environments the
   results of the model will apply (SPECIFICATION).

3) How to use the results of the model (STANDARDIZA-
   TION).

4) How can the results be shown to be "true", and how
   can the use of the results be evaluated (VERIFICA-
   TION and VALIDATION) (Ref 29:301).

Once the researcher has answered these questions and di-

vided the model into its elements, the computerization part

of the paradigm can begin. The output of analysis and meas-

urement should be in a form that can be programmed on a

computer. The model is executed by the computer program,

and the results are compared with the system and the con-

ceptual model. If the results are not what is expected, the

conceptual model is either re-examined or re-analyzed. If

the results are what is expected, the model can be used to

investigate the system.

Using this general model, a discussion of how the system

science paradigm is specifically applied to this research is presented in the following paragraphs.

The computerized scheduling of operations at a SMW can be thought of as a series of inputs, a process, and an output. The inputs are the operational resources of the SMW. The process is scheduling (both the computerized portion and the scheduler developed portion). The output of the process is the operations schedule for the month, along with accompanying statistics. A detailed look at the elements of the scheduling system and their environment is presented in Figure I-1. This model determines the analysis and measurements required to model the system.

This system can be thought of as taking the missile combat crews (MCCs) and allocating them to various jobs as specified by the heuristic algorithm used. The jobs are the activities that crew members must be scheduled for throughout the month. These activities include: alerts, training, leave, and other duties that preclude alert (such as meetings).

The simulation of the system takes a heuristic, or rule oriented, approach. This heuristic is implemented by a discrete event/network simulation model that handles the inputs to the system in a controlled fashion to generate a schedule. The crews are viewed as entities traveling through the network to different events. The activities that a crew must perform are assigned at the events. A heuristic algorithm is used to assign the crews to the
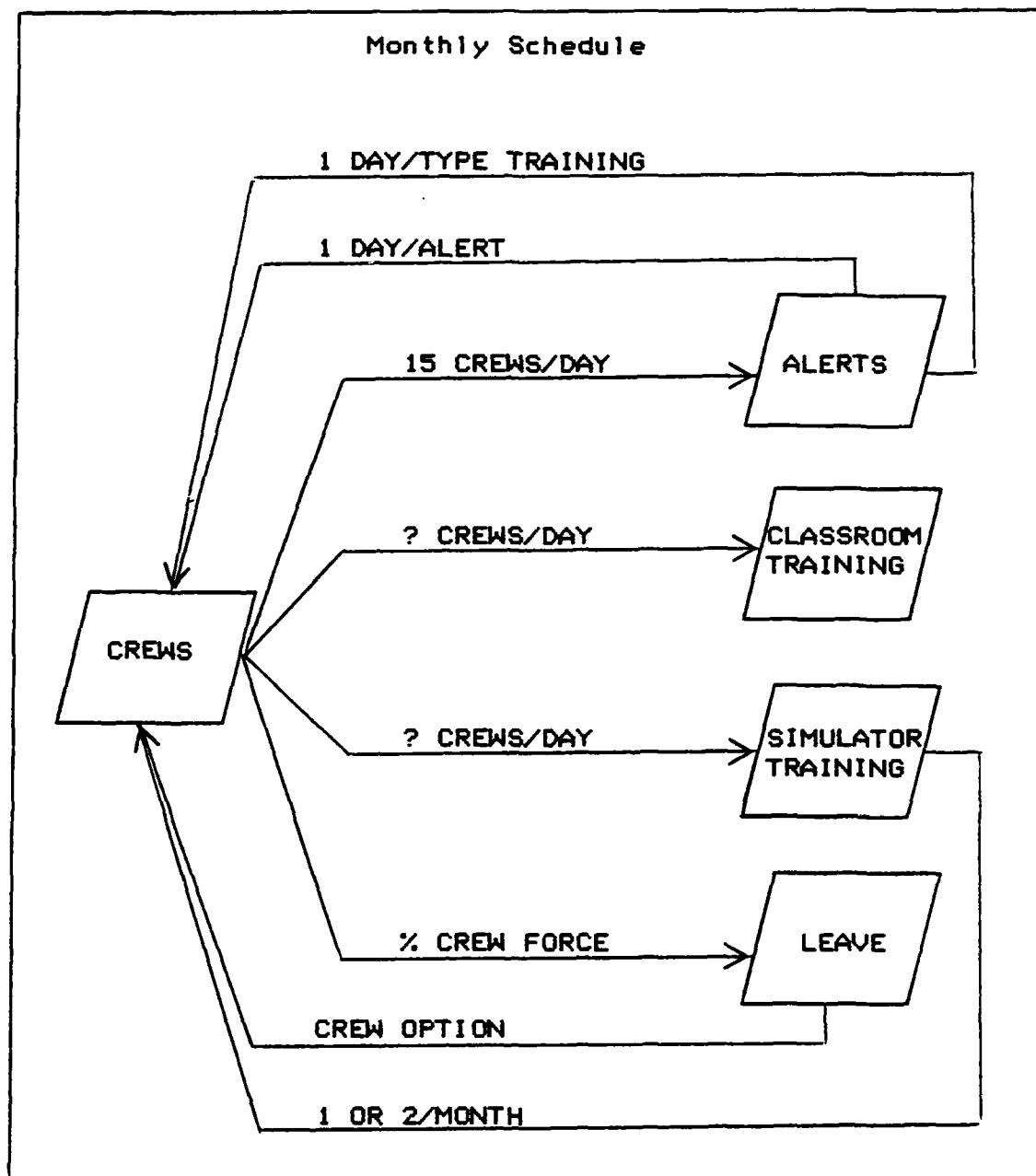
Figure I-1. Missile Scheduling System Structural Model

activities during the month. This model structure is the same as the system structure discussed in the preceding paragraph. The model and the system can be thought of as having a combined network/discrete event orientation.

Considering the four items that must be developed in the

analysis and measurement part (Language, Specification,

Standardization, and Validation/Verification),

a) The results of the simulation will be expressed in
   the form of a matrix representing the operations
   schedule for a month.

b) The results will apply to the operations portion at
   an SMW as long as the correct inputs have been spec-
   ified. This program will deal with a three-squadron
   wing, but the model can be changed to incorporate a
   four-squadron wing.

c) The results will be used to generate a set of alter-
   native schedules that can be compared. The best
   schedule, as judged by some criteria, can be
   implemented.

d) Verification and validation will be covered in depth
   in Chapter III of this research effort. The output
   of the model will be checked against the conceptual
   model to insure that the output is what is expected.
   Additionally, the model output will be compared with
   actual schedules to determine if the model produces
   an accurate operations schedule.

The model is built to run in the SLAM simulation lan-
guage on a Cyber (750 or 74) computer. The mathematical
model derived in the analysis and measurement part is
compatible with this language. A discussion of why this
particular model was chosen is appropriate at this time.

As has already been mentioned in the background portion

of this chapter, linear programming techniques form the majority of models used in this area. But there are several reasons to take a simulation approach rather than a mathematical approach for this study. A linear programming model for the scheduling problem would require hundreds of constraints and variables. Many aspects of scheduling would be difficult to put into a constraint format. The scheduling of leave is an example of this. Every leave for every crew would have to be considered a constraint. Any change in leave would change the entire model, because the constraints would change. In addition, any change to the input variables, such as a change in the number of crews, would require a complete revision of the program. A change in the number of crews would require the number of variables and the constraints to change. This makes a linear programming model large and inflexible to changes in the input variables. Linear programming problems of this size require a large amount of computer time to run. For example, one computerized model using an IBM 370 computer took up to 25 minutes of run time to develop the training portion of a schedule (Ref 8:46). This expense is difficult to justify to a decision maker. In addition, linear programming concepts are difficult to explain.

Ackoff (Ref 1:372-375) lists over a dozen reasons why computer simulation should be used. Several of these reasons that directly apply to the scheduling program are discussed in the following paragraphs.

Simulation requires the systematic gathering of data about a system and its environment. This leads to better understanding of the system by the designer, a better understanding of the model by the layman(decision maker), and can lead to insight into how to improve the process. Present schedules will be examined in detail to determine how schedules are constructed. These ideas will be used in developing the algorithms used in the scheduling program.

The information about variable relationships is easier to modify. Modifying inputs to the scheduling program does not require a complete modification of the program as a change to the constraints and variables of a LP program does.

The results of the process are easily explained and conveyed to the decision maker. The results of the scheduling program are placed into a scheduling matrix by using the SLAM language itself. A separate program would have to be built to do this for an LP program.

Separate subsystems of the process can be analyzed, and modified if desired. The various modules of the scheduling program can be changed without changing other program modules. This is not true of an LP program. Any change to a constraint or a variable may effect the entire model.

In this particular case, the model can be cheaper to run and faster to load. For example, changing the inputs on the scheduling program requires less time than re-programming a change to an LP program.

The model itself can be used to sell the program to the decision maker, because he can see it work. The scheduling program can be shown in a symbolic form that the decision maker can follow. In addition, a crew can be traced through the scheduling program to show how the program operates. This is not possible with an LP model.

The total system is investigated in a simulation approach reducing the chance of bias effecting the results. Separate LP models must be constructed for scheduling alerts, training, and leave. This can lead to bias, because the total system is not considered as a whole. The entire system is modeled at once in the scheduling program. (Ref 1:372-375)

In addition, there has been little research done in this area using simulation techniques. This research could lead to a better understanding of the role of simulation in scheduling support. This research can add to the knowledge of Decision Oriented Scheduling Systems (DOSS). The scheduling program is designed to fit into a DOSS network for missile scheduling. The intent of the research is to begin an exploration of the use of DOSS in this area of decision making, thus providing additional information on DOSS. For the above reasons simulation was chosen as the model for this project.

Two simulation languages were available to the authors for this research: SLAM and Q-GERT. Both of these languages have a network orientation like the scheduling sys-

tem. However, SLAM is the only language that can accomodate a combined network/discrete event orientation that is necessary in the scheduling program. SLAM is also more flexible than Q-GERT and there are specific subsystems in the model that are not feasible to program in Q-GERT. Both languages were compatible with the computer resources available. SLAM, however, is more widely used. Both languages are FORTRAN based. This gives program transportability, because FORTRAN is widely used in the military and civilian communities.

The program will use approximate 140,000 bytes of memory. This amount of memory is not compatible with a micro-computer, but the program can be executed on a mini-computer or a main frame.

SLAM

SLAM is the simulation language used in building the scheduling program. Background material on this language will aid in understanding the program. An excellent definition of SLAM is provided by A. Alan B. Pritsker in Introduction to Simulation and SLAM.

> SLAM, a new Simulation Language for Alternative Methods, is described in detail. SLAM is an advanced FORTRAN based language that allows simulation models to be built based on three different world views. It provides network symbols for building graphical models that are easily translated into input statements for direct computer processing. It contains subprograms that support both discrete event and continuous model developments, and specifies the organizational structure

I-30

for building such models. By combining network, discrete event, and continuous modeling capabilities, SLAM allows the systems analyst to develop models from a process-interaction, next-event, or activity scanning perspective. The interfaces between the modeling approaches are explicitly defined to allow new conceptual views of systems to be explored (Ref 27:vii).

The world view of the scheduling program is a combined network/discrete event orientation. The missile operations crews are thought of as transactions traveling through a network completing various events at different times during the month. FORTRAN subroutines are inserted into the SLAM program to direct crews to their next event or activity.

Once the model has been built, verified, and validated, a method can be developed to compare alternative schedules. Criteria for ranking these alternative schedules must also be developed. These criteria are applied to the alternatives to select the best schedule. The method used for comparing the alternatives must be flexible enough to allow a variety of criteria. This requirement is needed because every decision maker will have unique criteria by which he will judge the alternative schedules. The different criteria will effect which schedule is considered the best. A method of comparing alternative schedules is presented in Chapter IV of this report.

SUMMARY

Chapter I contains an overview of operational scheduling at strategic missile wings. How to computerize the sched-

uling system to improve the schedule is the purpose of this research effort. The chapter included a review of the resources available and the major activities that must be scheduled by the wing scheduler. In addition, a statement of the problem, a list of research objectives, a section on scope and limitations of the research effort, and background material, including previous research efforts in the area of scheduling were presented. Following these sections was a discussion of the general and applied methodology to be used in the research.

ORDER OF PRESENTATION

Chapter II will present a detailed look at the conceptual model used in building the computerized scheduling system. This model is broken down into subsystems, and each subsystem is discussed in depth.

Chapter III looks at model validation and verification. Each subsystem is checked against the conceptual model to see if its output agrees with what is expected. In addition, a schedule produced by the scheduling program is compared to an actual schedule from the 44th SMW.

Chapter IV develops an experiment investigating a method for comparing alternative schedules. MCDT and RSM techniques are used to determine the best scheduling policies. The experimental results are presented and discussed. Additionally, sensitivity analysis is performed on the model variables. The results of this analysis are also presented

and discussed.

Chapter V summarizes the research effort and presents
model results. Recommendations for further research are
discussed, and some concluding remarks about the research
effort are presented.

Two appendices are included with the report. Appendix A
contains the computer code and a user's guide for the sched-
uling program. Appendix B contains the data for the analy-
sis presented in Chapter IV.

CHAPTER II

THE MODEL

INTRODUCTION

A broad conceptual model for the scheduling program was presented in Figure I-1. A narrower and more defined conceptual model is present in Figure II-1 of this chapter. This model is then broken down into the lowest level of conceptual models, models of each program module. These outlines of the the program modules give a structure for analyzing and measuring the scheduling system. This fulfills part of step two of the system science paradigm presented in Chapter I. The remainder of this step is discussed in Chapter III when verification and validation are presented. These concepts presented for the program modules can be translated into computer algorithms for the computerization of the scheduling model, the third step of the system science paradigm.

An outline of the scheduling program is shown in Figure II-1. This outline shows tasks that must be accomplished by the scheduling program to simulate the system. This outline also applies to the scheduling system. The scheduler first gathers data that is needed to build the schedule. This data includes: the crew force size and structure, MPT availability, the number and size of classroom training,

```
┌─────────────────────────────────────────────────────────┐
│                   ┌───────────────┐                       │
│                   │ Input External│                       │
│                   │    Factors    │                       │
│                   └───────────────┘                       │
│                           │                               │
│                   ┌───────────────┐                       │
│                   │  Initialize   │                       │
│                   │   Program     │                       │
│                   └───────────────┘                       │
│                           │                               │
│                   ┌───────────────┐                       │
│                   │  Schedule     │                       │
│                   │ Leave/Alerts  │                       │
│                   └───────────────┘                       │
│                           │                               │
│               ┌─────────────────────┐                     │
│               │     Schedule        │                     │
│               │ Classroom Training  │                     │
│               └─────────────────────┘                     │
│                           │                               │
│                    ┌──────────────┐                       │
│                    │  Schedule    │                       │
│                    │    MPTs      │                       │
│                    └──────────────┘                       │
│                           │                               │
│                    ┌──────────────┐                       │
│                    │   Collect    │                       │
│                    │  Statistics  │                       │
│                    └──────────────┘                       │
│                           │                               │
│                 ┌──────────────────┐                      │
│                 │ Output Completed │                      │
│                 │     Program      │                      │
│                 └──────────────────┘                      │
└─────────────────────────────────────────────────────────┘
```
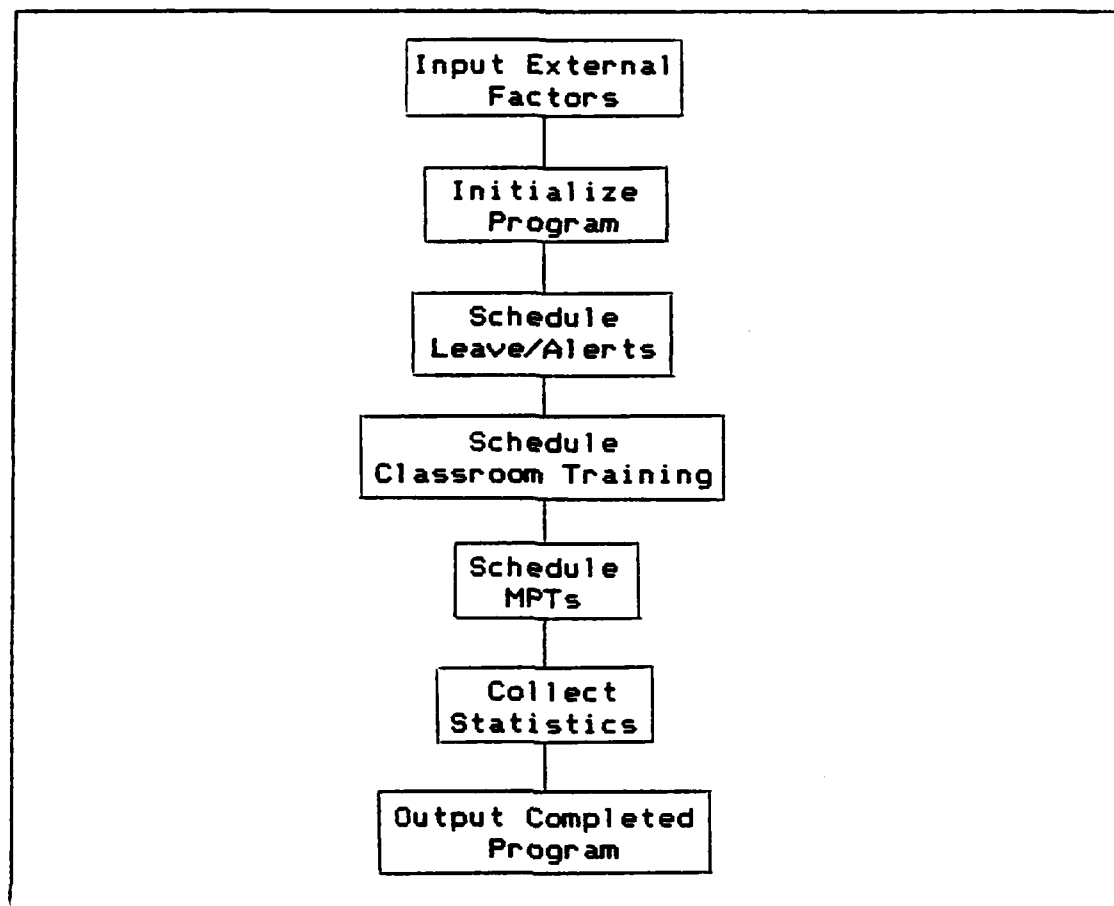
Figure II-1.   Crew Scheduling Program Modules

leave, and other duties. The scheduler schedules leave and
alerts first. These two activities are the most critical in
the schedule and all other activities are scheduled around
them. Classroom training and MPT sessions are assigned
next. Statistics for the schedule are calculated after it
is completed. Finally, the schedule is typed, printed, and
distributed. The scheduling program operates in this same
manner.

This model was designed as a set of modules using heur-
istic algorithms to build a missile operations schedule.

The model is flexible in that certain variables can be changed to modify the schedule that is built. This was done to allow the user to build a number of alternative schedules. An experiment is presented in Chapter IV that illustrates the ability of the model to construct alternative missile operations schedules and judge them based on a performance measure that the user chooses. The following is a list of the program modules:

    a. Initialization.

    b. Leave/Alert Scheduling.

    c. Classroom Training Scheduling.

    d. MPT Scheduling.

    e. Collecting Statistics.

    f. Output.

The following paragraphs will explain each module in greater detail discussing the objective, reasoning, and algorithm used for each module. An outline of each module is also presented. A detailed discussion of the computer code with program and variable listings is included in a user's guide contained in Appendix A of this report.

INITIALIZATION

    The objective of this program module is to prepare the program to build an operations schedule. The outline for the initialization module is shown in Figure II-2.

    The user must input the data on all MCCs into the program. This data includes:

```
┌─────────────────────────────────────────────────────┐
│              ┌─────────────────────────┐             │
│              │  Input Crew Information  │             │
│              └─────────────────────────┘             │
│                           │                          │
│             ┌──────────────────────────┐             │
│             │   Place First Day Crews   │             │
│             │         On Alert          │             │
│             └──────────────────────────┘             │
│                           │                          │
│            ┌────────────────────────────┐            │
│            │  Input Classroom Training   │            │
│            │        Information          │            │
│            └────────────────────────────┘            │
│                           │                          │
│               ┌────────────────────────┐             │
│               │    Input MPT Session    │             │
│               │      Information        │             │
│               └────────────────────────┘             │
│                           │                          │
│               ┌────────────────────────┐             │
│               │   Input Scheduling      │             │
│               │      Information        │             │
│               └────────────────────────┘             │
│                           │                          │
│               ┌────────────────────────┐             │
│               │   Begin First Crew      │             │
│               │     Leave Search        │             │
│               └────────────────────────┘             │
└─────────────────────────────────────────────────────┘
```
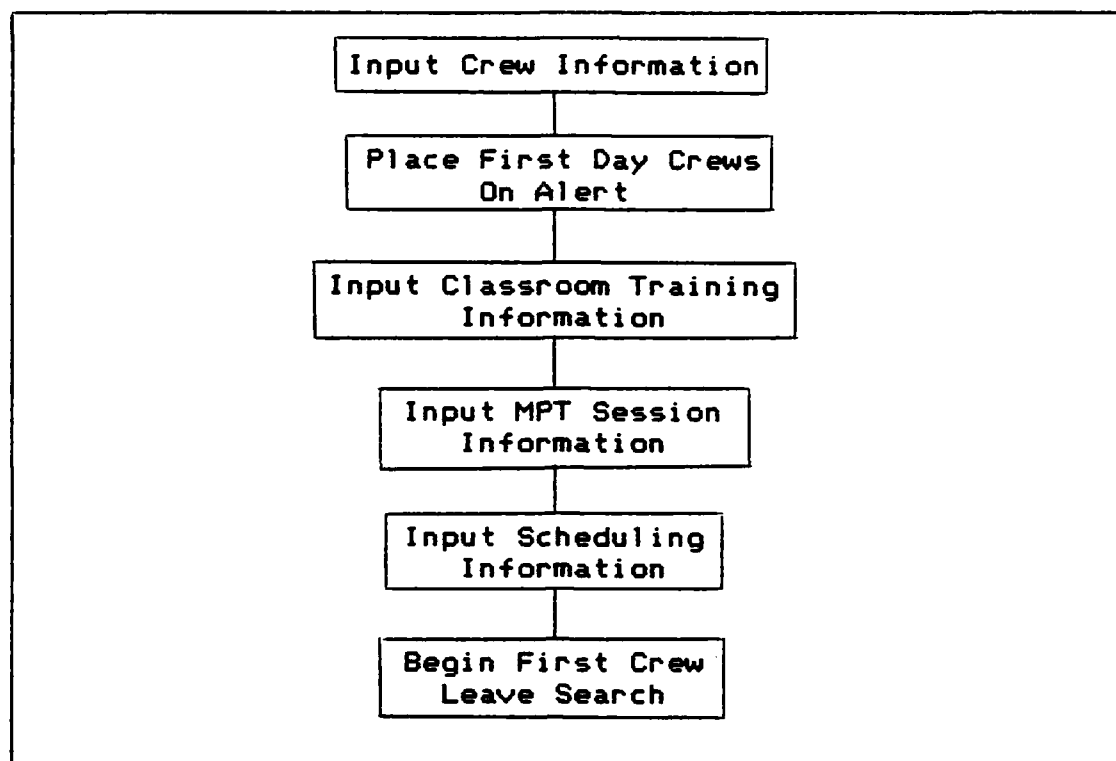
Figure II-2. The Initialization Module

    a.   Crew number.

    b.   Crew type.

    c.   Home LCC.

    d.   ACP/SCP qualification.

    e.   The beginning and ending dates of any leave.

Crews that have leave on the first day of the month are
placed on leave status. These crews will be placed in the
eligible-for-alert (EA) queue on the day that their leave
ends. This is done prior to placing crews in the EA queue,
because it is easier for the SLAM program to operate in this
manner.

The user must also input the 15 crews that have been
selected for alert on the first day of the month and which

II-4

LCC each crew is assigned to. All other crews are placed in
the EA queue and are eligible for alert.

Next, the data on the types of classroom training and
the days of the month that this training can be given are
entered into the program. The number of MPT sessions avail-
able on each day must also be input.

Finally, information on how to schedule events for the
crews must also be input by the user. This information
includes:

    a. The maximum number of alerts for each crew type.

    b. The priority rule for selecting each crew type from
        the EA queue.

    c. The number of days in the month.

    d. The number of crews in the crew force.

    e. The minimum and maximum number of students in each
        classroom training session.

Next, alerts and leave are scheduled for the entire crew
force for the month.

LEAVE AND ALERT SCHEDULING

The objective of this program module is to schedule all
leaves and alerts for the entire month for the crew force.
Any alerts that cannot be scheduled are identified and a
warning message is printed. The concept used in building
the code was to have the off-going crew trigger a search for
a new crew to be placed on alert at that LCC. The problem
with this concept arises when there are no crews in the EA

queue to be sent on alert on that day. If this occurs then no more alerts will be scheduled for that LCC because there are no more off-going crews. The program recognizes this problem and automatically triggers a search for a new crew for that LCC on the next day. Only one alert must then be manually scheduled by using this procedure. The program also outputs a warning message that no crew was scheduled for alert at that LCC on that particular day.

There are three possible activities that take place after a crew has been placed on alert (See Figure II-3). Activity one places the crew back in the EA queue after two days. The two day delay represents the day of return from alert plus 24 hours of crew rest.

Every crew is checked immediately after completing an alert to see if the crew has leave scheduled within the next two days. If leave is scheduled, then activity two is taken and that crew is placed in leave status. The crew is re-turned to the EA queue after the crew's leave is completed. The crew is placed on leave status when the leave period is scheduled to begin. If leave does not begin immediately after the alert, the crew is eligible to be selected for classroom training or an MPT session.

Activity three places the crew into the inactive queue. This queue is used to hold crews that have completed their maximum alerts for the month. The crew can still be sched-uled for classroom training and MPT sessions after they have been placed in this queue.
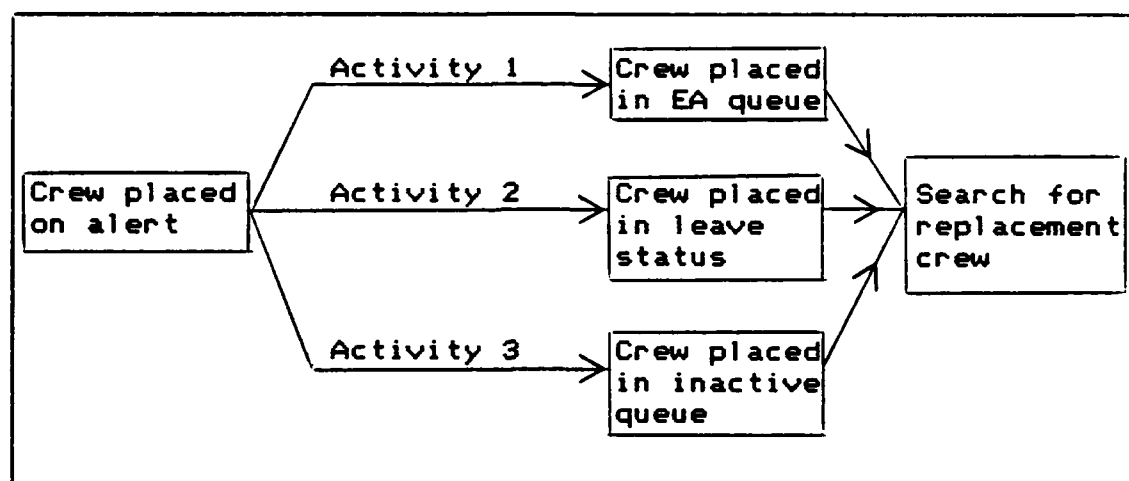
Figure II-3. The Alert and Leave Scheduling Module

All three activities trigger a search for a replacement crew at that LCC. This process is repeated for all 15 LCCs every day. The entire SLAM network involves this program module. This network and a description of it is contained in the user's guide in Appendix A.

The EA queue is searched every day, before alerts are assigned, for crews going on leave within the next two days. These crews are removed from the EA queue and placed on leave status. The crews are returned to the EA queue after their leave is completed. Crews entering the inactive queue are checked for any leave requests for the remaining days in the month.

Every crew can have up to three separate periods of leave specified for the month. The number of leave periods can be changed with minor program changes. Although the concept is called leave, this time off represents any activity that precludes the crew from performing alert duties.

An example of this is TDY, or upgrade training. The leave scheduling module is really a method of informing the program that any given crew is not eligible for alert for certain periods of the month.

CLASSROOM TRAINING

The objective of the classroom training module is to schedule all crews for the required classroom training for the month. Conceptually, the program tries to minimize the number of classes for each training type, and limit the number of crews in each class. These two objectives contribute to training effectiveness. The fewer the number of classes held, the more time instructors can spend on their other duties. The fewer crews per class, the more individual attention they can receive. The problem is that these two goals conflict, and some compromise between class size and number of classes must be found.

The outline for the classroom training module is shown in Figure II-4. The model uses a two-pass search system to schedule classroom training. For each training type, the first available day is checked to see if there are enough crews to have a class. The range of class sizes can be set by the user. The initial class can range from the minimum value set by the user to 80 percent of the maximum value. If all crews have not been scheduled for training after the first pass then the number of crews allowed in each class is increased to the maximum value set by the user, and the

```
        ┌──────────────────────┐
   ┌───>│  Search for training │
   │    │type and available days│
   │    └──────────────────────┘
   │                 │
   │    ┌──────────────────────┐
 No └────│ Are there enough crews?│
        └──────────────────────┘
                     │
                    Yes
                     │
                     ▼
             ┌──────────────┐
             │ Schedule crews│
             └──────────────┘
                     │
             ┌──────────────────┐
             │  Print type training│
             │   and class size │
             └──────────────────┘
                     │
             ┌────────────────────┐
             │ Print unscheduled crews│
             └────────────────────┘
```
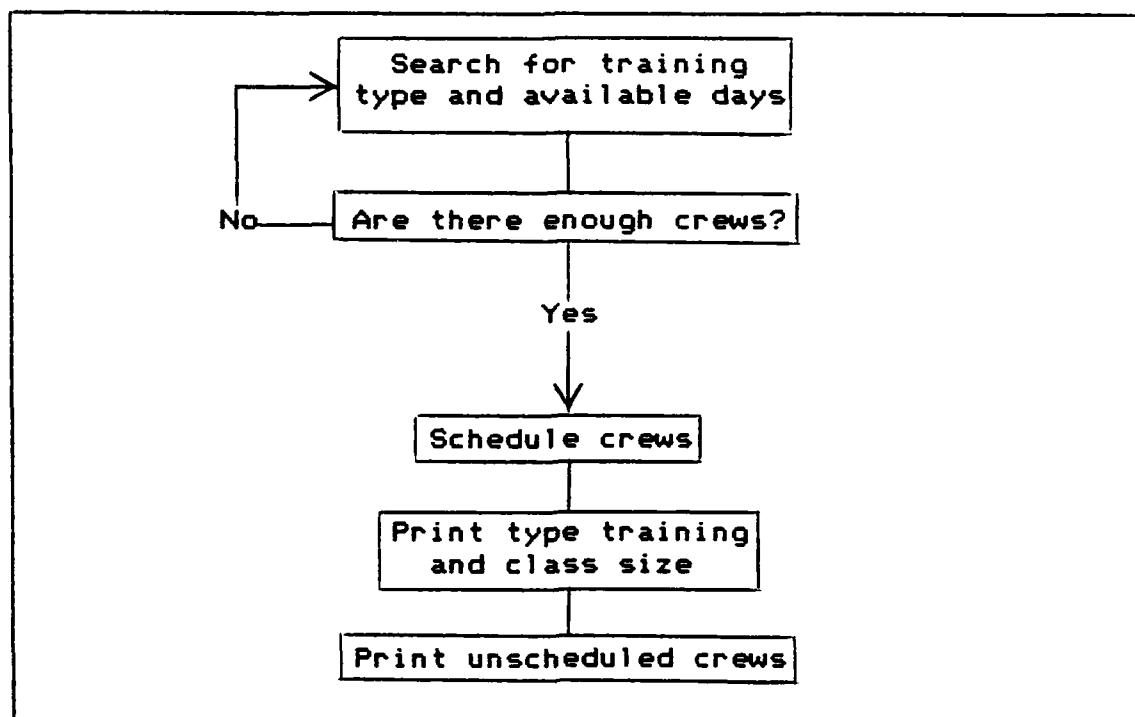
Figure II-4. The Classroom Training Module

search is repeated (the second pass). The first pass checks

available training days in ascending order while the second

pass checks training days in descending order. This heur-

istic evens out the class sizes throughout the month and

increases the number of crews scheduled for training. Any

crew that is not scheduled for training after the second

pass is flagged by a warning message. Finally, the number

of classes and the class sizes for each type of training are

printed. This routine is only effective at scheduling the

entire crew force for an activity. Crews should be sched-

uled for individual activities using the leave module. This

subroutine does find a compromise between the class size and

the number of classes presented. The values can be changed

by varying the class size range or number of available training days.

MPT TRAINING

The objective of this module is to schedule crews for MPT sessions. The number of MPT sessions for each crew in a month is determined by the user. Additionally, the number of days between MPT sessions can be set. Currently, the model is set to schedule every crew for one MPT session per month. If more than one MPT session per crew is set, the user must supply the minimum number of days between MPT sessions for the same crew. The outline for the MPT training module is presented in Figure II-5.

This search is a one pass system that is executed N times, where N is the number of MPT sessions for each crew. The program checks the number of MPT sessions available every day. If sessions are available, crews are scheduled to fill the slots. The crews are checked in descending order by crew number. A crew that has been scheduled for that round of MPT sessions is not considered. The number of MPT sessions available is decreased for a given day as crews are assigned. The next MPT round is scheduled in the same manner, except a check is made to see if the crew had an MPT session within the specified minimum number of days. If this is the case, then the crew is not assigned and the next available day is checked. Any crew that has not been as-signed to the appropriate number of MPT sessions is flagged

```
          ┌──────────────┐
          │ Search days for │
  ┌──────→│ MPT availability │
  │        └──────────────┘
  │               │
  │               ↓
  │        ┌──────────────────────────┐
No├────────│ Are MPT sessions available? │
  │        └──────────────────────────┘
                    │
                   Yes
                    │
                    ↓
          ┌──────────────────────────┐
          │ Schedule crews and decrease │
          │      available sessions      │
          └──────────────────────────┘
                    │
                    ↓
          ┌──────────────────────────┐
          │   Print unscheduled crews   │
          └──────────────────────────┘
```
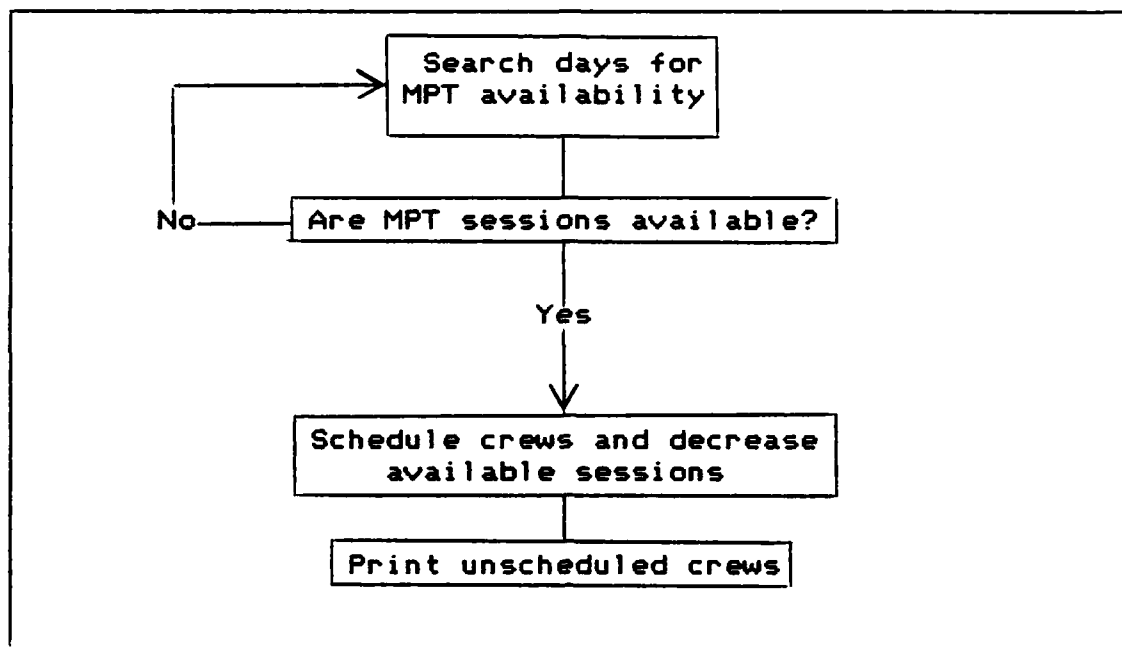
Figure II-5.   The MPT Training Module

by a warning message. This routine is similar to the
classroom training module in that it can only be used to
schedule the entire crew force for an activity.

COLLECT STATISTICS

     The objective of this module is to calculate any statis-
tics desired about the monthly schedule. The statistics
calculated in this module are necessary to compare alterna-
tive schedules. This information is used in the experiment
discussed in Chapter IV.

     Information is extracted from the completed schedule.
The statistics are then calculated from this information.
The outline for the statistics module is shown in Figure
II-6. The program has already been provided with a crew

```
┌─────────────────────────────────────────────────────────┐
│              ┌──────────────────────────┐                │
│              │     Search Schedule       │                │
│              │   for crew information    │                │
│              └──────────────────────────┘                │
│                           │                               │
│            ┌──────────────────────────────┐              │
│            │  Compute mean and variance    │              │
│            │        for line crews         │              │
│            └──────────────────────────────┘              │
└─────────────────────────────────────────────────────────┘
```
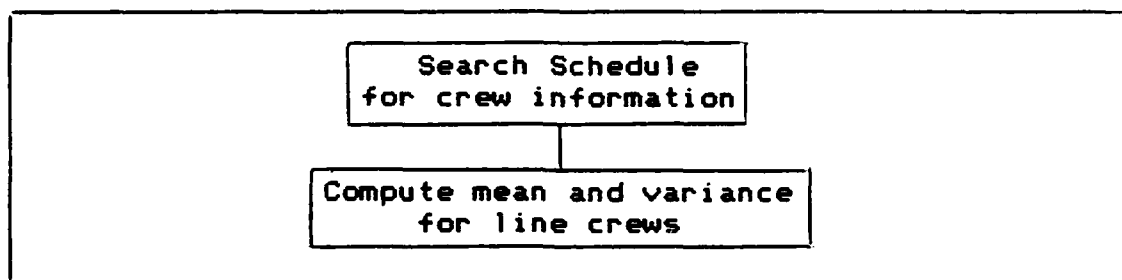
Figure II-6. The Statistics Module

type (line, flight commander, or DOT/DOV) for every crew
listed in the schedule. Only line crew data is used to
compute mean and variance statistics for alert rate,
integral alert rate, and number of free days. It is the
information on line crews that is important in determining
statistics for the schedule, because DOT/DOV and flight
commander crews have a *fixed number* of alerts per month.

OUTPUT

The objective of the output module is to present the
needed information to the user in a readable format. The
schedule is presented in a coded form. For example, the
value 1 on a given day indicates that a crew has an alert at
LCC 1. A complete listing of the codes is contained in the
user's guide (Appendix A). The monthly statistics printed
about the schedule include:

    a. Crew Number.

    b. Number of alerts for each crew.

    c. Number of integral alerts for each crew.

    d. The total number of free days for each crew.

e. The cumulative line crew statistics.

The outline for the output module is presented in Figure II-7. In addition, sample output from the program is included in Figure II-8 at the end of this chapter.

PROGRAM STRUCTURE

The conceptual modules for the scheduling program were translated into computer code as subroutines. These subroutines were added to the SLAM executive program to form the total scheduling program. The following paragraphs are a brief introduction to each program subroutine. A complete listing and discussion of each subroutine is contained in the user's guide in Appendix A of this report. The following is a discussion of the scheduling program subroutines.

The initialization subroutine prepares the program to build a schedule and gathers the data needed for program execution. The event subroutine contains the logic that switches the leave and alert subroutines on and off as required. The leave and alert subroutines schedule crews for leave and alert for the entire month. The MPT subroutine schedules crews for MPT sessions while the classroom training subroutine schedules classroom training. Finally, the statistics subroutine calculates the statistics needed for schedule evaluation and the output subroutine prints out the completed schedule and statistics. Also, warning messages are printed by the respective subroutine for any activity not scheduled. For example, a crew missing an MPT session

```
┌─────────────────────────────────────────────────┐
│         ┌───────────────────────────┐            │
│         │    Print Crew Schedule     │           │
│         └─────────────┬─────────────┘            │
│          ┌────────────┴──────────────┐           │
│          │  Print monthly statistics  │          │
│          └────────────┬──────────────┘           │
│        ┌──────────────┴────────────────┐         │
│        │  Print line crew statistics    │        │
│        └───────────────────────────────┘         │
└─────────────────────────────────────────────────┘
```
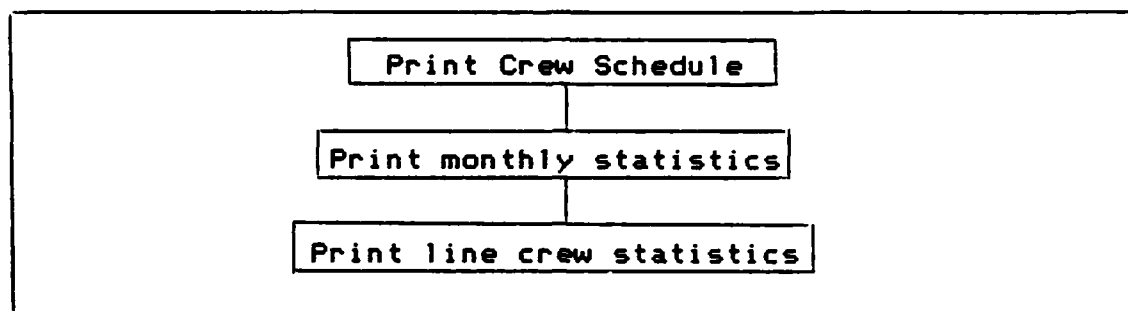
Figure II-7. The Output Module

would cause a warning message to be printed by the MPT subroutine.

SUMMARY

The scheduling program was designed in a number of modules. These modules are translated into subroutines in the computerized model.

The model is designed to provide a monthly missile operations schedule for a strategic missile wing with three operational squadrons. The model is a combined discrete event/network model using heuristic algorithms to develop a schedule. Missile operations crews can be thought of as transactions flowing through a network that complete events at specified times. After the model is built, the next step in the design process is to verify and validate the model design. The model's ability to build a usable operations schedule is tested in this step. Verification and validation, the remaining portions of step two of the system science paradigm, are covered in Chapter III.

<u>Sample Warning Messages:</u>

  <u>Alert:</u>
    crew file empty on day 26 for lcc 12
    no scp crew for scp 1 on day 28

  <u>Training:</u>
    crew 35 not scheduled for XX training
    crew 86 not scheduled for mpt 2

<u>Training Recap:</u>

    for XX training on day 8, 10 crews scheduled
    .
    .
    .

<u>Schedule Example:</u>

    crew scheduling matrix

```
         1    2    3    4  .   .   . 29   30   31
   1     1   99    0    0  .   .   . 11   99    0
   2     0   XX    1   99  .   .   .  0    6   99
   .
   .
   .
  90     0   XX   11   99  .   .   .  1   99    0
```

<u>Statistics:</u>

  <u>Individual:</u>
    crew 1, alerts 4, integral alerts 3, free days 14, actual number of leave days 0
    .
    .
    .

  <u>Cumulative Line Crew:</u>
    number of alerts:
    mean = 6.444   standard deviation = .50156

    number of integral alerts:
    mean = 4.093   standard deviation = .85271

    number of free days:
    mean = 14.389   standard deviation = .76273

(<u>NOTE:</u> XX indicates user supplied numeric code.)

Figure II-8. Sample Scheduling Program Output

CHAPTER III

VERIFICATION AND VALIDATION

INTRODUCTION

The scheduling program must be verified and validated before it can be used. These verification and validation processes are the final step in the system science paradigm used in the design. The verification and validation processes are iterative, involving a re-conceptualization and re-computerization if the model cannot be verified and validated in its present form. This chapter contains background material on verification and validation and outlines a methodology for performing these procedures for the scheduling program. This methodology is then applied to the model and the results are discussed.

VERIFICATION

Verification is the process of checking the computer code to insure that the program executes as intended (Ref 27:10). The program was built in modules to simplify the debugging and verification efforts. Each module was checked separately in the scheduling program by printing the output of the module. This output was then verified to insure the program had functioned properly before adding the next module. The verification process is discussed for each module

III-1

in turn in the following paragraphs.

The initialization module was first checked to insure that the user inputs were placed in the correct storage locations in the program. This included both crew and training information. Secondly, the module was checked for its ability to place crews on leave status correctly. This involved insuring that the starting and ending dates of all leaves scheduled to begin on the first day of the month were correct. The alert and leave scheduling module schedules crews for leave after the first day of the month. Finally, the module was checked to insure that the 15 crews selected to perform alert on the first day of the month were scheduled correctly. The alert and leave scheduling module performs this function after the first day.

The alert and leave scheduling module was checked next. First, the program's ability to fill alert requirements for the month was checked. Using the computed schedule, each day was checked to insure a crew of the proper type (only ACP/SCP-qualified crews to ACP/SCPs) was sent to each launch control center (LCC). Next, by using an infeasible crew force size, the program-generated warning messages for unfilled alerts were also tested. The ability of the module to select next alert crews based on a user specified percentage was then demonstrated. Additionally, the schedule was checked to verify crews were placed in the inactive queue after completing their maximum number of alerts for the month.

Finally, the leave function of the alert and leave scheduling module was verified. Again, using the computed schedule, each leave request was compared to the schedule to insure proper beginning and ending leave dates. The EA queue was also checked to insure the return of crews following leave completion. Proper leave updates to the schedule were also checked for crews entering the inactive queue.

The classroom training module was checked to make sure that all crews were scheduled for classroom training correctly. The schedule was used to verify each crew was either scheduled for each type of training or was reported as not being scheduled in a warning message. In addition, the ability of the module to stay within the given class size ranges was checked.

The MPT training module was verified in the same manner as the classroom training module. In addition, the ability to separate MPT sessions by some minimum number of days was checked for proper operation.

The statistics module was verified by hand calculating the individual crew and cumulative line crew statistics. Once again the computed schedule was used to gather the necessary information.

The operation of the output module was checked throughout the verification process. As each module was verified, the printed information concerning that module was checked for errors.

VALIDATION

Validation is the process of insuring that the scheduling model accurately reflects the scheduling system to the degree desired. The ability of the scheduling program to build a feasible operations schedule is the accuracy needed. The output from the scheduling model is validated by comparing it to existing operations schedules from an operational SMW. The inputs from these manually generated schedules are used as inputs for the scheduling program. Validation was accomplished by taking a missile combat crew schedule and duplicating it. This was done by determining the inputs to the scheduling program based on the actual schedule. The two schedules were then compared using criteria discussed in the following paragraphs. If the schedules did not agree then the scheduling program was adjusted and executed again until it closely matched the existing schedule.

The following output information is compared to the data from the manually generated schedules:

a. Is the schedule feasible? (does it meet all constraints).

b. The alert rate and variance.

c. The average number of free days.

d. The number of class days for each type of classroom training.

e. The class size and variance for each type of classroom training.

f. The average number of MPT sessions for the crew

force.

g. The integral alert rate and variance.

This information is important because it captures the essence of what an operations schedule should do. The schedule must be feasible, or it is worthless. To be feasible the schedule must fill the alert requirements which means that there will be an alert rate and an average number of free days for the crew force. The class size, variance and number of classes is a measure of the schedule's ability to efficiently allocate the crews to classroom training sessions. The average number of MPT sessions is also a measure of this ability. The integral alert rate is a measure of the scheduling program's ability to adjust to an existing schedule. Can the program be tuned to develop a given integral alert rate, is the question addressed by this criterion. Significant differences between the scheduling model and the manually generated schedules are discussed in the next section.

SCHEDULE SELECTION

A schedule from Ellsworth AFB was selected to validate the scheduling program. This schedule was selected from among schedules from Malmstrom, F.E. Warren, and Whiteman AFBs. The schedule from Ellsworth AFB was selected because it typifies a schedule from a three-squadron strategic missile wing (SMW). The schedule from Malmstrom AFB was not chosen because it contained a large number of crew changes

spread throughout the month. A crew change is a crew split where the crew partners are reassigned to other crews or other duties. There were over 40 such changes in the schedule from Malmstrom AFB. The scheduling program is not designed to handle this type of problem. Every SMW will have some crew changes in a month. If these crew changes are made at the beginning of the month, for example, the changes would pose no problem for the program. (A methodology for dealing with crew changes is discussed later in this chapter.) An integral crew structure was assumed as one of the limitations of this research effort. With this number of crew changes, the scheduling program would have to run on a person by person not a crew basis. A problem significantly more difficult to solve than the one for crew scheduling.

The schedule from F.E. Warren was not selected because this SMW has four squadrons. The program is designed for a three-squadron SMW. A four squadron scheduling program can be built with modifications to the existing program. These modifications would be major in terms of the amount of code changed, but not in terms of the conceptual model. Whiteman AFB schedules were not selected, because the Emergency Rocket Communications System (ERCS) is located at this base. The schedule here requires special rules and procedures to schedule crew members associated with the ERCS system. The program should be easy to modify to consider these rules. With these problems in mind, the Ellsworth schedule was selected as the best choice for validation.

The problem of split crews still had to be addressed
with the Ellsworth schedule, but on a smaller scale.
Ellsworth, like other SMWs, has several reserve crew members
(RSCMs). An RSCM is one who is currently without a crew
partner. How to input the RSCM into the scheduling program
was the initial problem faced in the validation process.
There are two types of RSCMs encountered in the Ellsworth
schedule. An RSCM that is crewed with a permanent partner
some time during the month is the first type of RSCM.

This crew was treated as being unavailable for alert (on
leave) until the crew was formed. The second type of RSCM
is a crew member that is not crewed with any one crew member
for the entire month. This type of RSCM was ignored in the
initial validation computer run. This means that several
crew members capable of performing alert were not included
in the schedule. The consequence of ignoring these crew
members will be discussed in the initial validation run
results.

INPUT PREPARATION

The next problem is to determine the scheduling inputs
for the program by looking at the manually generated sched-
ule. The exact inputs to this schedule are unknown and had
to be estimated. The following variables were set based on
the manually generated schedule for the first validation
run:

a. The 15 crews for alert on the first day.

III-7

b. All leave requests for the crews.

c. The actual training days used for classroom train-
    ing in the Ellsworth schedule.

d. The actual number of MPT slots in the Ellsworth
    schedule.

The Ellsworth schedule was set to give all line crews and
flight commander crews approximately one MPT session per
month. Only 24 percent of DOT/DOV crews, however, are
scheduled for an MPT session in the Ellsworth schedule. The
scheduling program cannot set one MPT rate for one type of
crew and a different rate for another crew type, therefore,
an MPT rate of one MPT per crew for all crews was set. The
ability to set different MPT rates for different crew types
is not needed. If one crew type is to receive a different
number of MPT sessions, these sessions can be manually
reconciled.

    Scheduled duties and meetings are treated as leave for
the crew even if only one crew partner is scheduled for the
meeting. This same policy is adopted for only one crew
partner taking leave. The crew partner not on leave can be
scheduled for other duties manually. Integral leave is the
policy at SMWs, but there are exceptions.

    In addition, the class size range is set between six and
fifteen crews per class. The maximum number of alerts for
line, flight commander, and DOT/DOV crews is eight, four,
and two respectively.

    There are also five internal percentages in the sched-

uling program that effect the selection of crews for the next alert. The first three percentages affect the next alert at non-SCP LCCs while the last two affect SCPs. These are:

1. The percentage of time that a DOT/DOV crew is selected for a non-SCP alert.

2. The percentage of time that a flight commander crew is selected for alert at their home LCC.

3. The percentage of time a crew is selected for alert at their home LCC.

4. The percentage of time a DOT/DOV crew is selected for alert at an SCP.

5. The percentage of time an SCP crew is selected for alert at their home LCC.

VALIDATION RUN RESULTS

Table III-1 contains the results for the first validation run of the scheduling program compared to the Ellsworth schedule. The schedules are compared using the procedure discussed in the validation section of this chapter.

The alert rates from the two schedules are within three percent of one another, but the scheduling program has a higher variance. The 10 alerts performed by the reserve crew members, who were never crewed with a permanent crew partner, is the probable reason for the difference in alert rates. These crews were never considered in the initial

Table III-1. First Run Results

	Ellsworth Schedule	Scheduling Program
ALERTS		
Mean	6.29	6.48
Variance	1.08	1.56
INTEGRAL ALERTS		
Mean	3.94	2.79
Variance	1.67	2.79
FREE DAYS		
Mean	12.98	12.65
Variance	11.11	24.01
MPT		
Mean	0.74	0.93
CLASS SIZE		
Class 1		
Mean	9.30	12.86
Variance	9.12	2.48
Class 2		
Mean	9.4	11.86
Variance	5.82	8.48
CLASS NUMBER		
Class 1	10	7
Class 2	10	7

run, decreasing the size of the crew force. A smaller crew force will have a higher alert rate and a lower number of free days. Changing the crew force to include the reserve crew members should decrease the alert and increase the number of free days for the automated schedule.

The integral alert rate was higher and the variance lower for the Ellsworth schedule compared to the scheduling program. The integral alert rate for the program was set at 50 percent, while the Ellsworth schedule's integral alert rate was actually 63 percent. Setting the integral alert rate higher in the program should increase this statistic. A thorough study of the integral alert rate is presented in the experiment discussed in Chapter IV.

The program did not schedule six crews for an MPT session. These crews can be substituted manually for DOT/DOV crews in a short period of time. This is not a serious situation. An increase in MPT availability on a certain day or an increase in the total number of MPT slots should solve the problem.

The class size ranges for both schedules are comparable. The scheduling program used a class size range of between six and fifteen crews per class. The Ellsworth schedule class size ranges were from six to sixteen crews and five to thirteen crews for type one and type two training, respectively. The scheduling program uses fewer days for training and has a better overall variance because of the heuristics employed in the program to minimize class days. Seven crews

could not be scheduled for type two training by the program. A large amount of leave by these crews is the reason for this. This is not an unusual situation at SMWs. Several crews must be given individual classroom training every month because of leave considerations. These crews can be manually scheduled for type two training by using one of the following methods:

a. Schedule the crew for individual crew training at some time during the month.

b. Schedule the crew on a training day that increases the class size past the maximum value of 15 crews per class.

c. Switch the day for type one training so that type two training can be given on that day.

The scheduling program is given eight days as possible training days for classroom training. The program used seven of these days. The Ellsworth schedule was originally thought to have eight training days scheduled, but actually had ten days of classroom training. The scheduling program was given fewer days in which to schedule training and used fewer days with a more constant class size.

Because of the differences in the two schedules it was decided to perform another validation run with the following two changes. First, increase the percenatge of time a DOT/DOV crew is selected for an SCP alert. This change should increase the alert rate for DOT/DOV crews to their maximum of two (Three DOT/DOV crews performed only one alert

in validation run one.). Also, two crews should be added to
the crew force to simulate the unaccounted for RSCMs of the
first validation run. Since the Ellsworth RSCMs performed
ten alerts, each added crew in the scheduling program was
given leave for part of the month to decrease their alerts
to ten also.

The integral alert rate percentage was not changed in
the program to see what effect the other changes would have
on the schedule. Table III-2 presents the data from the
second validation run compared to the Ellsworth schedule.

The changes to the program inputs resulted in several
significant changes in the schedule. First, the alert rate
and variance in the alert rate decreased while the integral
alert rate remained the same. This increased the average
number of free days as expected. The addition of the two
crews caused the changes in the schedule.

The second validation run showed that the scheduling
program is capable of developing a feasible schedule given
realistic inputs. The model can accurately reflect a typi-
cal missile operations schedule and, therefore, appears val-
id for its intended use. One additional validation run was
performed to see the difference that occurs if the integral
alert rate is increased from 50 to 63 percent and 10 train-
ing days are available for scheduling classroom training.
This validation run was made to see if the scheduling pro-
gram could build a better schedule than the manually gener-
ated schedule in terms of the validation criteria. The

Table III-2. Second Run Results

	Ellsworth Schedule	Scheduling Program
ALERTS		
Mean	6.29	6.23
Variance	1.08	1.30
INTEGRAL ALERTS		
Mean	3.94	2.82
Variance	1.67	2.69
FREE DAYS		
Mean	12.98	13.21
Variance	11.11	15.37
MPT		
Mean	0.74	0.93
CLASS SIZE		
Class 1		
Mean	9.30	13.14
Variance	9.12	4.48
Class 2		
Mean	9.4	11.86
Variance	5.82	13.14
CLASS NUMBER		
Class 1	10	7
Class 2	10	7

results are listed in Table III-3.

Changing the integral alert percentage in the program
from 50 to 63 percent increased the integral alert rate to
57 percent as compared to 63 percent for the Ellsworth
schedule. This is a significant increase and demonstrates
the usefulness of such a variable in the program.

Increasing the number of days available for each type of
classroom training also caused a significant change. The
number of days when classroom training was scheduled in-
creased to eight, and only three crews were not scheduled
for classroom training. All of these crews can be manually
scheduled for training.

Both reserve crew members were checked to insure that
another crew member was available to perform alerts with
them when their alerts were scheduled during the month.
This check was made manually, and the crew partners were as-
signed manually. This process does not take a significant
amount of time to perform.

The changes to the inputs for the scheduling program
further improve the schedule. These changes show that the
schedule can be tuned to achieve a desired scheduling ob-
jective. This process will be studied in detail in the
experiment presented in Chapter IV.

Several problems were noted in the validation process.
The technique of adding crews to simulate the existence of
reserve crew members is a useful one. This technique can be
used whenever a reserve crew member is available in the

Table III-3. Third Run Results

	Ellsworth Schedule	Scheduling Program
ALERTS		
Mean	6.29	6.23
Variance	1.08	1.35
INTEGRAL ALERTS		
Mean	3.94	3.52
Variance	1.67	2.99
FREE DAYS		
Mean	12.98	13.18
Variance	11.11	15.13
MPT		
Mean	0.74	0.92
CLASS SIZE		
Class 1		
Mean	9.30	11.39
Variance	9.12	9.57
Class 2		
Mean	9.4	11.25
Variance	5.82	5.64
CLASS NUMBER		
Class 1	10	8
Class 2	10	8

scheduling program. The program has a definite handicap in dealing with split crews, however. In order for the scheduler to use the program effectively, crew changes will have to be kept to a minimum. Ideally, crew changes would be made once a month (preferably at the beginning of the month) and input to the initial data inserted into the program. This technique can be applied to crew changes, but it is of limited use in making a large number of them.

Another problem area is the number of scheduled duties precluding alert that are scheduled. The three periods of leave set in the scheduling program are sometimes not enough to schedule all activities for the crew force. The number of leave periods should be increased to at least four or five periods per crew.

Chapter IV will use the verified and validated scheduling program to analyze program variables and to present a procedure for comparing the worth of alternative schedules. Such a procedure is valuable to a scheduler because he can generate several schedules for use in a given month. His selection can be based on the use of the procedure outlined in the next chapter. This procedure is one of many such methods that can be used. The discussion in Chapter IV provides an example of the type of research that can be done with the scheduling program.

EXPERIMENTAL DESIGN AND APPLICATION

INTRODUCTION

Now that a program has been built that schedules missile operations crews, the model can be used to accomplish the objectives of this research effort. Those objectives are the development of high quality missile operations schedules, and the performance of analysis that supports this development. The problem that now faces any prospective user of the program is the definition of a "high quality" schedule. The solution to this problem cannot be unique. Every decision maker will have a slightly different opinion about what constitutes a good schedule. What the scheduling program will allow a decision maker to do, if the program is combined with some performance measure, is to build a schedule that meets the specifications of a given decision maker. The objective of the experiment discussed in this chapter is to provide a decision maker with a method of determining the quality of a schedule. This method does not guarantee that a schedule produced by the program is an "optimal" schedule. Rather the method introduced in this chapter provides the decision maker with a method of comparing alternative schedules and deciding which of these alternatives he prefers.

Two things must be designed to perform the experiment:

a method of scoring alternative schedules, and a method to use for finding the best alternative.

The performance measure used in this experiment is developed by using a technique from the field of Multi-Criteria Decision Theory (MCDT). It must be emphasized that the performance measure derived here is not the only one that can be built. There are many different techniques that can be used, and this is an example of one of these techniques. Any performance measure can be used in this experimental design if the scheduling program can compute the necessary data for the performance measure. The particular technique demonstrated in this experiment is worth assessment. This MCDT technique is simple to use and provides a numerical score for each alternative.

Using an MCDT technique is applicable to this problem because there are multiple objectives. Multi-Criteria Decision Theory is a set of analysis techniques that are used to simultaneously meet several objectives.

After the performance measure has been developed, an experimental design must be built for testing the various alternatives and finding the best schedule. Response surface methodology was used as the design, and this process is discussed after the section on worth assessment.

A discussion of the experiment and the results of the experiment follows. This discussion will include a step by step analysis of the experimental procedure and the building of the performance measure. Finally, sensitivity analyses

will be performed on the results of the experiment to see how sensitive the alternative selected is to changing factors in the scheduling model and to changing performance measures.

WORTH ASSESSMENT

Worth assessment is used to compare alternative schedules. This technique develops a measure of performance for a given schedule. The scheduling program can then build a number of alternative schedules that can be evaluated in a structured manner to find the best alternative. A set of objectives that describes the performance of a schedule must be developed first. From these objectives, a performance measure can be built.

Worth assessment is the MCDT technique used in this research effort, because the technique aids in determining what the decision maker feels is important about the schedule. Worth assessment is being used as an example of one type of performance measure. The technique of worth assessment can be adapted to any decision maker, to develop a performance measure specifically tailored for him. A discussion of the worth assessment technique follows.

Worth assessment, as outlined by DeWispelare (Ref 13), has six steps:

1. List the overall objectives or attributes.

2. Construct a hierarchy of these performance criteria.

3. Select physical measures for the lowest level attri-

butes.

4. Define the relationship between the attributes and
   the physical measures.

5. Establish the relative importance of each attribute.

6. Adjust the weights of the attributes to increase
   the decision maker's confidence in their accuracy.

Worth assessment assumes that the attributes are "worth
independent". Worth independence is defined as the decision
maker being willing to get more of one attribute for less of
another with no change in the decision maker's satisfaction.
The satisfaction that a DM feels for a particular attribute
value is the worth of that attribute at that value. The re-
lationship between the worth of an attribute and the minimum
and maximum values of that attribute is assumed to be linear
in the procedure.

Building an objectives hierarchy is the first step in
the Worth Assessment procedure. This hierarchy is con-
structed starting with an overall objective for the sched-
ule. This primary objective is broken down into a series of
sub-objectives with the lowest level sub-objectives, or at-
tributes, being measureable. These lowest level attributes
will be used to score schedules that are developed. This
objectives hierarchy must agree with the decision maker's
concept of what attributes are important to the schedule.
If the objectives hierarchy does not agree with the decision
maker's ideas then the hierarchy is re-designed until it
agrees with the desires of the decision maker.

Angell (Ref 2) discusses the need for the development of such an objectives hierarchy. The hierarchy that Angell derives in his report serves the same purpose as the one in this report; to provide a measure of the worth of a schedule and to allow comparison of alternative schedules.

An objectives hierarchy for the scheduling program is presented in Figure IV-1. This hierarchy is similar to Angell's (Ref 2:27), but the hierarchy addressed here is focused specifically on the issues developed by the scheduling program, not simply on scheduling in general.

The six measureable attributes developed for the scheduling program are: MPT time, classroom effectiveness, maximum alerts, difference in alert rates, free days, and integral alert rate. These attributes are measureable in that each one has a measure of effectiveness (MOE) associated with it. The MOE is some physical measure of each attribute. The measure can be related to how well the attribute has been fulfilled by a particular alternate schedule. The MOEs for each attribute are discussed in the following paragraphs.

MPT time will be measured by the number of MPT sessions that each crew receives during the month. The decision makers at a strategic missile wing are usually very interested in this measure, and, generally, have some amount of MPT time that they desire to see each crew receive. The number of MPT sessions was chosen as the MOE for this attribute for this reason.

MAINTAIN CREDIBLE
DETERRENT CAPABILITY

TRAINING
EFFECTIVENESS

RESOURCE
UTILIZATION

MORALE

MPT TIME

CLASSROOM
EFFECTIVENESS

ALERT
RATE

ALERT
VARIANCE

INTEGRAL
ALERTS

FREE
DAYS
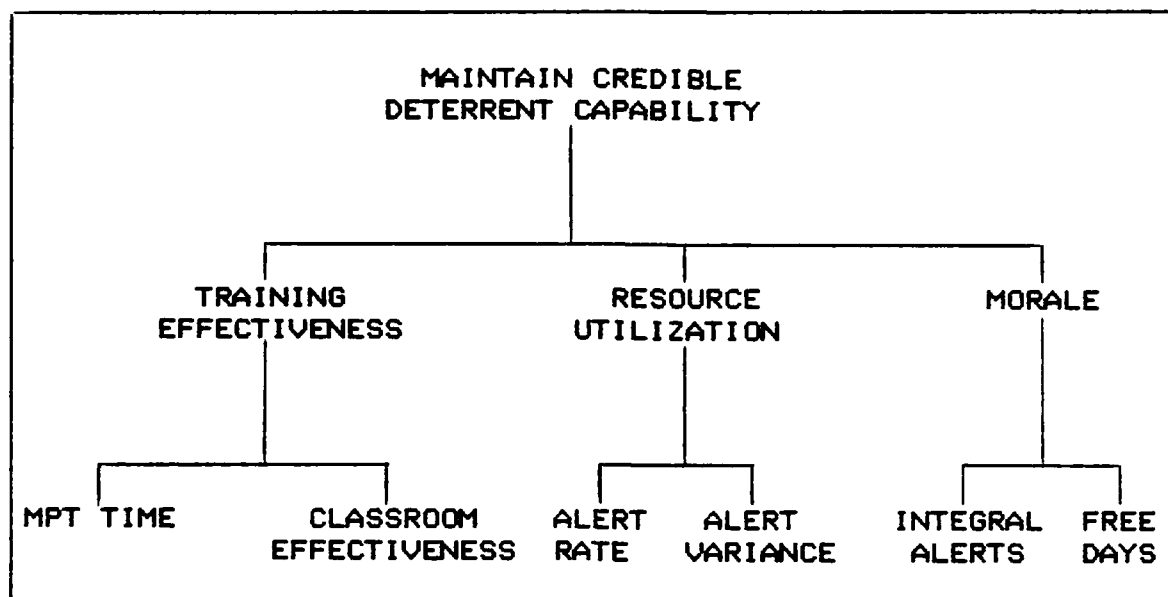
Figure IV-1.   Objectives Hierarchy for Operations Schedule

Classroom effectiveness will be judged by the variance in the class sizes of the different classroom training types. Initially, the MOE for this attribute was the number of training days for each training type divided by the total number of training days available. It was discovered in the validation portion of the research that the number of training days used by the program varied only slightly making this a poor MOE for this attribute. Although the quality of training is effected by many factors, class size is the only factor effected by the scheduling program. A constant class size certainly increases the probability that all crews are being trained to the same level of knowledge and performance, therefore variance in class size was selected as the MOE for this attribute.

The average alert rate and variance in alert rate are

chosen as MOEs for maximum alerts and difference in alert
rates respectively. These two measures appear to fit the
attributes well.

The average number of free days, and the percent of in-
tegral alerts performed were chosen as MOEs for the last two
attributes for the same reason.

There are several considerations taken into account when
determining the relationship between the physical measure
and the attribute. The following is a list of the consider-
ations, suggested by DeWispelare:

   a. Is the measure continuous or discrete?

   b. Does the measure possess upper and lower bounds?

   c. What values of the measure can be associated with
       a worth of zero and one?

   d. Does a higher value of the measure show a greater
       or lesser worth?

The objective of these questions is to determine the
relationship between the MOE and the worth of the attribute.
In other words, a method of converting the MOE value to a
worth value is defined by these questions.

The relationship between MOE and worth values is assumed
to be linear in the worth assessment procedure. Once the
quesions outlined above have been answered, converting an
MOE value to a worth value is a matter of determining the
maximum and minimum values for the MOE, determining the MOE
value for a particular schedule, and performing linear
interpolation to detemine the worth value for the particular

attribute in question. The worth value of an attribute is
assumed to be a value between zero and one. Table IV-1 sum-
marizes the relationship of MOE values to worth values for
the scheduling program objectives hierarchy.

Many of the values in Table IV-1 must be computed based
on the size and structure of the crew force. An example of
the worth assessment procedure will be shown to illustrate
how these values can be computed.

The next step in the worth assessment process is to
establish the relative weights of the objectives and attri-
butes in the hierarchy. To do this the attributes are di-
vided into groups with the relative importance of the attri-
butes within each group being established first. The objec-
tives and attributes for the scheduling program are divided
into five groups: the primary objective, the three sub-
objectives beneath the primary objective, and the six lower
level attributes which are divided into three groups of two
attributes each.

DeWispelare suggests one method of establishing the
relative importance of the attributes within each group.
His method is to have the decision maker rank the attributes
on a scale from one to one thousand. The reason for using
such a large scale is that some decision makers have diffi-
culty trying to distinguish between attributes on a smaller
scale (zero to one, for example). It is important to note
that the weights of the attributes within each group must
sum to 1000 to allow this method to work. The rank of each

Table IV-1. MOE Characteristics for Each Attribute

MEASURE	1	2	3	4	5	6
DISCRETE						
CONTINUOUS	X	X	X	X	X	X
UPPER BOUND	X	X	X	X	1.0	X
LOWER BOUND	0	0	X	0	0	0
WORTH=0	0 MPT	X	X	X	0	0
WORTH=1	X	0	X	0	1.0	X
HIGH VALUE IS WORTH MORE	X				X	X
LOW VALUE IS WORTH MORE		X	X	X		

X Answer must be determined analytically based on
the structure of the crew force.

attribute is divided by 1000 after the decision maker has
completed his ordering of all groups to provide attribute
weights with values between zero and one.

Once the relative importance of each attribute within
the group has been determined the relative weight of any
attribute compared to the entire performance measure can be
found. The weight of any attribute is determined by multi-
plying the relative weight of the attribute by the relative
weights of all objectives that it is a sub-objective of. An
example of this process should make this statement clear.

Assume that a decision maker has been asked to establish

the relative importance of the attributes and objectives

listed in Figure IV-1. These attributes are first divided

into five groups composed of the following attributes:

   a. Group 1--Maintain a Credible Deterrent Capability.

   b. Group 2--Training Effectiveness, Resource Utiliza-
       tion, and Morale.

   c. Group 3--MPT time and Classroom Effectiveness.

   d. Group 4--Maximum alerts and difference in alert
       rates.

   e. Group 5--Free days and integral alerts.

The relative importance of the attributes within each group

are determined next.

   Maintaining a credible deterrent capability is given a

rank of 1000 because it is the only objective in this group.

Assume that the ranks of the three sub-objectives in group

two are given ranks of 400, 400, and 200 respectively. Ad-

ditionally, assume that the following ranks are given to the

six lower level attributes in groups three, four, and five:

   a. Group 3: MPT time--600

               Classroom Effectiveness--400

   b. Group 4: Maximum alerts--500

               Difference in alert rates--500

   c. Group 5: Free Days--600

               Integral Alerts--400

Note that the ranking of each attribute must be divided by

1000 to find the relative weight. This is shown in Figure

IV-2.

```
                    MAINTAIN CREDIBLE
               DETERRENT CAPABILITY (1.0)



       TRAINING (.40)      RESOURCE (.40)        MORALE
       EFFECTIVENESS       UTILIZATION            (.20)


  MPT TIME        CLASSROOM      ALERT    ALERT    INTEGRAL   FREE
   (.60)        EFFECTIVENESS    RATE    VARIANCE   ALERTS    DAYS
                    (.40)       (.50)     (.50)     (.40)    (.60)
```
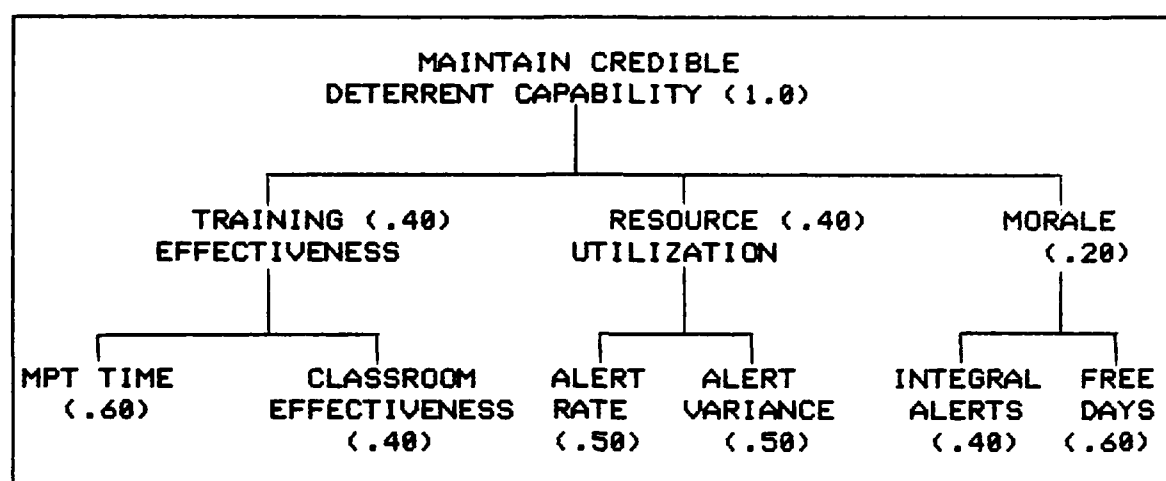
Figure IV-2. Attribute Weights for Objectives Hierarchy

To determine the weight of an attribute in comparison to
the entire performance measure, the following method is il-
lustrated using the attribute of maximum alerts. The rela-
tive weight of the maximum alerts attribute (0.5) is multi-
plied by the relative weight of the resource utilization
sub-objective (0.4) (maximum alerts is a sub-objective of
resource utilization) and by the relative weight of the pri-
mary objective (1.0) (resource utlization is a sub-objective
of the primary cbjective). The weight of the maximum alerts
attribute is found to be 0.2. This value indicates that 20
percent of the overall performance measure for a schedule is
determined by the maximum alerts attribute. All of the oth-
er attribute weights can be determined in the same manner.

The weighting scheme is shown to the decision maker, and
can be adjusted if the decision maker feels that the weights
do not accurately reflect his attitudes.

With the weighted hierarchy completed, alternative

schedules can be scored using the performance measure. The highest score is the best alternative schedule. As an example, assume a missile wing has 75 line crews, 13 DOT/DOV crews, and 15 flight commander crews. Additionally, assume that the scheduling program generated a schedule that contained the following data:

1. Average number of MPT sessions = 2.0.

2. Average variance in class size = 4.5.

3. Average number of alerts = 6.5.

4. Variance in alert rate = 1.5.

5. Average number of free days = 12.0.

6. Percentage of integral alerts = 75%.

First, the worth of each MOE value must be calculated. The relationship between the maximum and minimum values of the MOE and the worth of the attribute is assumed to be linear.

The worth of the measure for MPT time as listed in Table IV-1 is 1.0 for an average of two MPT sessions per crew.

The minimum value of for the variance in class sizes is zero while the maximum value must be calculated analytically. If it is assumed that the class sizes range between 10 and 15 crews and that a maximum of 10 days of classes can be offered, then the maximum variance is 10.5. Given an actual value for the variance of 4.5, the worth of this attribute is .57.

The upper bound for the alert rate was assumed to be 8 alerts per crew for this example, but the lower bound must

be calculated analytically. Assuming a 30 day month and
that DOT/DOV crews perform 2 alerts per month and flight
commander crews perform 4 alerts per month, the 75 line
crews must perform 364 alerts to insure that all launch
control centers are manned every day. This is equivalent to
an alert rate of 4.85 alerts per month, the lower bound.
Again, using linear interpolation, the worth of this measure
is .48.

The lower bound for the variance in alert rates is zero,
by definition, and the upper bound must be calculated. The
maximum variance can be found if the minimum number of crews
perform all the alerts, and the rest of the crew force per-
form zero alerts. This variance assumes that the DOT/DOV
and flight commander crews perform their usual number of
alerts per month. If the 75 line crews must perform 364
alerts, then these alerts can be handled by 45 line crews
performing 8 alerts, and one additional crew performing 4
alerts. The maximum variance based on this information is
14.49. With an actual variance of 1.5, the worth of this
attribute is 0.9.

Free days are defined as days where the crew has no
scheduled duty or is on leave. The minimum number of free
days is 0, and the maximum number must be calculated ana-
lytically. Given that the minimum alert rate is 4.85 alerts
per month, the number of free days will be 30 minus twice
the alert rate (count each alert as two days, one day for
the alert and one day for return) plus four training days

(two MPT days and two classroom training days). The maximum number of free days is 16.3 days. With an actual value of 12.0 days, the interpolated worth value is .74.

The upper and lower bounds for integral alerts are the same as the worth value upper and lower bounds, that is, zero and one respectively. The worth of the measure, for this example, is .75.

Once the worth values for the six lower level attributes have been determined, the performance measure for the schedule can be found. This value is found by multiplying the weight of the attributes by the calculated worth values and summing these values.

As an example, if the weight of the attribute MPT time (0.24) is multiplied by its worth value (1.0) to determine the performance measure value for this one attribute. ~ this case, the value is 0.24. The same calculation can be done for the other attributes to determine their performance measure values. These values for the other attributes include:

    a.   Classroom effectiveness—0.0912.

    b.   Maximum alerts—0.096.

    c.   Difference in alert rates—0.18.

    d.   Free days—0.0888.

    e.   Integral alerts—0.06.

The performance measure values for each of the attributes is summed to determine the performance measure for the schedule. For the example, this value is 0.756.

The use of the worth assessment technique provides a method of comparing different scheduling alternatives. The problem with using the method is that worth independence of the attributes must be assumed or proven. Proving worth independence can be a time consuming and difficult procedure. For this reason, more research must be done in the area of performance measures for evaluating scheduling alternatives. This example shows one possible procedure for deriving a performance measure, there are many others. Sensitivity analysis will be done in the experiment to find how the performance measure varies with changes in the values used in the worth assessment procedure. These values include: the weights of the attributes, the crew force structure, the upper and lower bounds of the MOEs, and the additional user inputs to the scheduling program.

A HEURISTIC APPROACH

Response surface methodology (RSM) will be used to analyze the set of alternative schedules in this experiment. RSM is an analysis tool that finds the best levels of a given set of variables, called factors, that create the best yield. In this case the yield is the performance measure. The approach outlined in the following paragraphs was adopted from Myers (Ref 24).

This experiment lends itself to RSM techniques because it contains a number of factors (the variables in the scheduling program) that affect a yield (the performance

measure). Additionally, the relationship between the fac-

tors and the yield is unknown and, possibly, complex. This

relationship will have to be estimated, which makes the use

of techniques such as mathematical programming difficult.

RSM graphically depicts the relationship of the factors to

the yield. Mathematical programming is a more difficult

technique to describe to a layman (such as a decision

maker), and the relationship of the variables to the

performance measure is not as easy to see. Finally, RSM

should give the researchers a better understanding of the

process that builds the schedule.

The output from the scheduling program is a missile

operations schedule plus statistics about that schedule.

This output is used as the input to the performance measure

equation. The independent variables in the scheduling pro-

gram that influence the schedule must be determined. These

variables are the factors, and the yield is the performance

measure. For the scheduling program the independent varia-

bles include:

    a. Factors internal to the program:

        1. The percentage of time that a DOT/DOV crew is

            selected for the next non-SCP alert.

        2. The percentage of time that a flight commander

            crew is selected for the next non-scp alert.

        3. The percentage of time that a crew is selected

            for the next alert to their home LCC.

        4. The percentage of time that a DOT/DOV crew is

selected for an SCP alert.

 5. The percentage of time an SCP crew is selected
for the next SCP alert at their home LCC.

   b. Factors external to the program:

      1. The number of ACP/SCP qualified crews.

      2. The total number of crews in the crew force.

      3. The number of days that a crew cannot be sched-
uled for the month.

   c. Optional factors that affect the schedule:

      1. The priority rule for selecting crews from the
eligible for alert queue.

      2. The maximum number of alerts that each crew
type can perform.

      3. The maximum and minimum class size for class-
room training.

The factors are divided into three groups because the
number of factors is too great to depict graphically if
handled as one group. The second and third group of factors
will be held fixed during the RSM experiment and are dis-
cussed in the sensitivity analysis portion of this chapter.
The first group will be tested in the experiment to deter-
mine the best internal factors for the performance measure.
The factors internal to the model represent ways in which
the model itself can be changed to improve the quality of
the schedule. Factors external to the model represent a
means of determining the best crew force structure for a
given performance measure. The optional factors represent

ways that the decision maker can improve the quality of the schedule through implementable policies.

The experimental design for this research effort was originally based on the use of multivariate regression techniques. It was discovered that the internal factors of the model were not useable with regression analysis, however. The internal factors can be broken into two groups: factors one, two, and three deal with selecting the next crew for non-SCP alerts, and factors four and five are used in selecting crews for SCP alerts. The factor values in each of these groups cannot add to any value greater than one. There are many values within the factor space that each factor is not allowed to take for this reason, making the variables discontinuous. Continuity is a vital assumption in regression analysis and designing regression experiments would be difficult with the limitations on values that the variables can take. An alternate experimental design had to be developed.

The revised approach is a heuristic based on graphing the response surface with a series of data points gathered throughout the factor space. Linear interpolation is used to fill in the details of the response surface that have not been specifically determined.

The three major factors will be graphed to provide an idea of the best area on the response surface for operations. The major factors are factors one, two, and three (the percentage of time that a DOT/DOV crew, integral flight
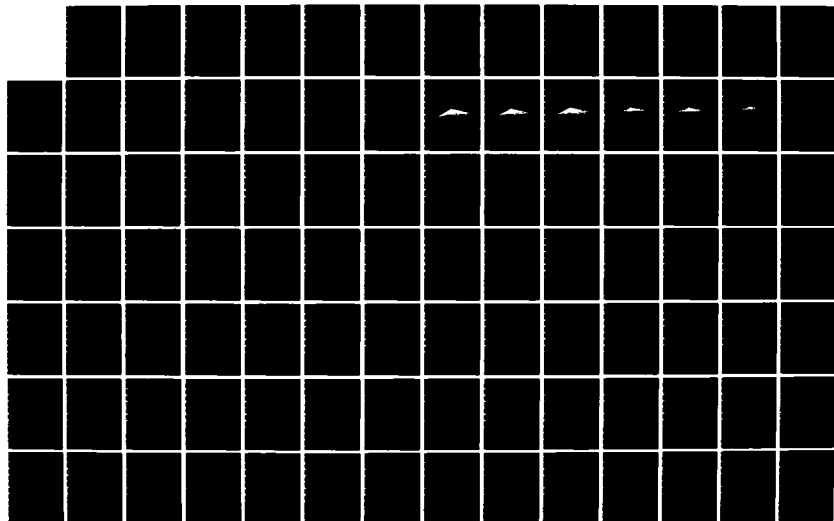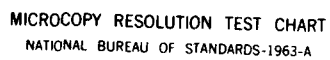
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

comander crew, and integral crew are selected for the next
non-SCP alert). These factors contribute more to the per-
formance measure value than factors four and five (the per-
centage of time a DOT/DOV crew and integral crew are se-
lected for the next SCP alert) because they contribute to
selecting 80 percent of the alerts. These factors are
plotted on two separate graphs, two factors versus the
performance measure value on each graph. The areas on the
graphs where the performance measure values are the highest
are looked at to determine a best factor combination of the
three major factors.

Factor combinations in these smaller areas are then
closely investigated to determine the best factor combina-
tions. Several trials are used to determine the value of
each data point. The means and variances of the data points
are computed and compared to determine the best factor set-
tings.

The best settings of factors four and five are found by
varying the levels of these factors using ten data points
from the first experiment. The means and variances of these
data points will be calculated to determine the best factor
settings for factors four and five. The best factor setting
based on the alternative schedules examined can be found in
this manner.

THE EXPERIMENT

The performance measure used in the experiment is the

same performance measure outlined earlier in this chapter. The objectives hierarchy and attribute weights are shown in Figure IV-3.

This performance measure was developed as one representative performance measure. Two other possible measures are checked in the sensitivity analysis portion of this chapter. Several inputs normally supplied to the program by the scheduler were fixed for the purposes of the experiment. The crew force size was fixed at 90 crews consisting of 15 flight commander crews, 21 DOT/DOV crews, and 54 line crews. These crews can perform a maximum of four, two, and eight alerts, respectively. This crew force structure was selected because it is typical of all three-squadron SMWs.

The class size range for classroom training was set between six and fifteen crews per class with a maximum of twelve training days available for each training type. Two types of classroom training and one MPT session were given to each crew every month. These values are all representative of normal values for these inputs.

Additionally, the EA queue used the fewest number of alerts performed to date as a priority rule for searching the file for the next crew to perform alert. The external factors for this experiment were set to a maximum crew force (90 crews) and a minimum number of SCP crews (39). This crew force structure is also typical for three-squadron SMWs.

A total of 51 data points was gathered throughout the

```
                    MAINTAIN CREDIBLE
                DETERRENT CAPABILITY (1.0)
                            |
        ┌───────────────────┴───────────────────┐
   TRAINING (.40)        RESOURCE (.40)        MORALE
   EFFECTIVENESS         UTILIZATION           (.20)
        │                     │                   │
   ┌────┴────┐          ┌─────┴─────┐        ┌─────┴─────┐
MPT TIME   CLASSROOM    ALERT     ALERT    INTEGRAL   FREE
 (.60)    EFFECTIVENESS RATE    VARIANCE    ALERTS    DAYS
           (.40)        (.50)    (.50)      (.40)     (.60)
```
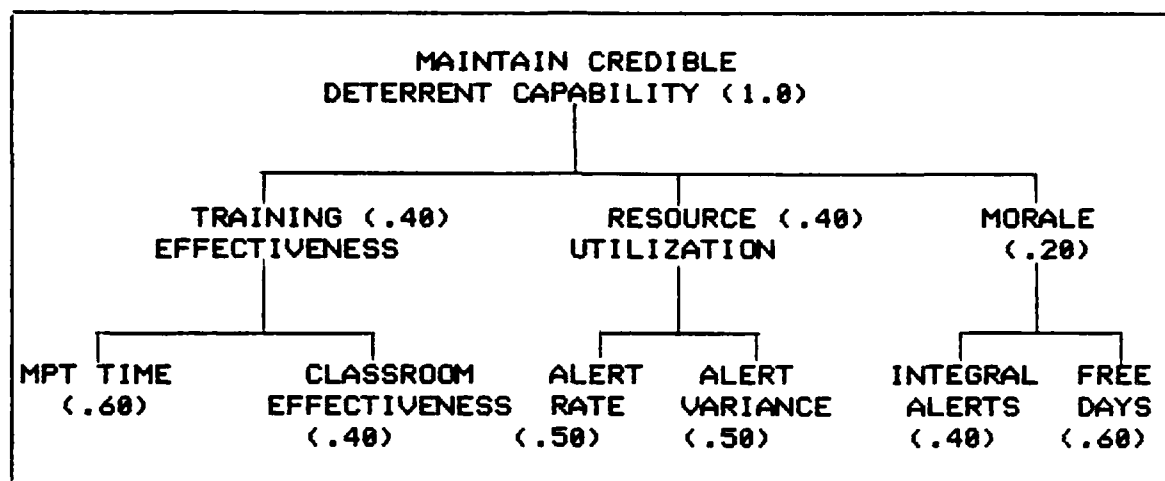
Figure IV-3. Objectives Hierarchy for The Experiment

factor space for graphing the response surface. At least
four trials of each data point were made. The average
performance measure value for each data point was computed
from the trials and used in the plotting of the repsonse
surface.

Factors one, two, and three were graphed separately
against the performance measure first to see if any one
factor appeared to be linearly related to the performance
measure. These graphs are in the back of this chapter in
Figures IV-5 through IV-7.

The graphs show the performance measure is directly
proportional to factor three and inversely proportional to
factors one and two. The graphs also indicate that the best
area for the performance measure appears to be between 0.0
and 0.3 for factor one; 0.0 and 0.2 for factor two; and 0.6
and 1.0 for factor three.

From this information, it was decided that factors one

and two would be graphed against factor three and the performance measure to determine the shape of the response surface and the location of the best factor area.

The graphs for the response surface were plotted on a Hewlitt Packard HP7228A plotter using the S statistical language and a VAX 11-780 computer. The graphs are plotted in a three-dimensional perspective format. The graphs for the repsonse surface are included at the end of this chapter.

Factors four and five were checked against ten different data points from the experiment. Five levels of these two factors were originally run for each data point. The results showed that the best factor levels for factors four and five were between 0.1 to 0.3 and 0.7 to 0.9 respectively. Two additional data runs were made for each of the ten data points to see if there was a single best factor level for each of these two factors in this area.

EXPERIMENTAL RESULTS

Ten data points were checked in the area of highest performance measure values to determine what the best factor combinations for factors one, two, and three were. The data points were picked as a representative cross section of the nineteen data points in the area of interest. These data points are listed in Appendix B. Two data points were found to be better than all of the others in the area. The statistical tests used to compare the data points are also con-

tained in Appendix B. The variances of the data points were
first tested for equality with an F-test. If the variances
were equal, a T-test was used to test the means of the per-
formance measure values. A Wilcoxon-Mann-Whitney rank sum
test was used to test the means of the values if the vari-
ances were found to be different. The factor combinations
of 0.1, 0.1, 0.8 and 0.2, 0, 0.8 for internal factors one,
two, and three, respectively, were found to produce the best
schedules in terms of performance measure values. The data
points, means, and variances are summarized in Table IV-2.

Factors four and five were compared in a similar manner
and two factor settings were found to be better than all
others for these factors. Factor values of .1 and .2 for
factor four and .9 and .8 for factor five were determined to
produce the best schedules. These data points and their
means and variances are summarized in Table IV-3.

What these factor settings mean are that setting the
program to select a DOT/DOV crew for the next non-SCP alert
ten percent of the time, selecting a flight commander for
alert at his home LCC ten percent of the time, selecting an
integral crew for the next alert eighty percent of the time;
selecting a DOT/DOV crew for the next SCP alert ten percent
of the time, and selecting an integral crew for the next SCP
alert ninety percent of the time produces the best schedule
for this performance measure. Setting these factors to
levels of twenty, zero, eighty; twenty and eighty percent,
respectively, produces an equally good schedule.

Table IV-2. Best Data Points for Factors
            One, Two, and Three for Performance
            Measure One

Number of	Data Points			Measure	
Observations	F1	F2	F3	Mean	Variance
10	0.1	0.1	0.6	0.954	4.87E-5
10	0.1	0.3	0.6	0.955	2.82E-5
10	0.1	0.2	0.7	0.958	4.09E-5
14	0.1	0.1	0.8	0.964	2.42E-5
10	0.1	0.0	0.9	0.933	4.75E-5
18	0.2	0.2	0.6	0.958	6.19E-5
12	0.2	0.1	0.7	0.960	7.30E-5
10	0.2	0.0	0.8	0.961	9.85E-5
10	0.3	0.1	0.6	0.959	9.54E-6
12	0.3	0.0	0.7	0.958	3.35E-5

The other factor combinations in this area also produce
reasonable schedules. That is, the performance measure
values for these factor combinations are almost as good as
the two best factor combinations. This means that the op-
timal area for this performance measure is relatively flat
as can be seen from the response surfaces depicted in
Figures IV-8 and IV-9.

The internal factors only affect the integral alert rate
of the schedule. This is true until the DOT/DOV crews stop
performing their maximum number of alerts. This event oc-
curs when internal factor one is set too low. Factor one

Table IV-3. Factors Four and Five Values Used
in Determining the Best Levels of These
Factors for Performance Measure One

Number of Observations	Factor Levels		Performance Values	
	4	5	Mean	Variance
10	0.1	0.9	0.953	2.30E-5
10	0.2	0.8	0.956	7.16E-5
10	0.3	0.7	0.951	7.70E-5
10	0.4	0.6	0.943	4.50E-5
10	0.6	0.4	0.944	7.18E-5
10	0.8	0.2	0.942	6.30E-5
10	1.0	0.0	0.943	9.00E-5

must be set high enough to insure that the DOT/DOV crews

perform at least 42 alerts per month. This number of alerts

is roughly equal to nine percent of the total alert load

that must be performed. If the DOT/DOV crews do not perform

their 42 alerts, the alert rate and the variance in alert

rate for the line crews increase; the number of free days

also decreases. This causes the performance measure value

for the schedule to decrease. The results of the analysis

also indicate that the variance in the model is small with

little change occuring over a broad area of the response

surface. This result can be seen by examining the variances

listed in Tables IV-2 and IV-3.

Finally, the results of the experiments show that sched-

uling alternatives can be compared by the method outlined in

this chapter. The best internal factor setting for the
alternative schedules examined can be found using this
method for a given performance measure.

SENSITIVITY ANALYSIS

Sensitivity analysis of the results obtained in this
experiment are considered next. This analysis will look at
several program variables. The effect on internal factor
settings using different performance measures will be dis-
cussed first. The effect of having the entire crew force
ACP/SCP qualified will be presented next. The effect on the
performance measure of adding an assumed leave distribution
to the schedule inputs will be looked at third. The effect
of changing the priority rule for searching the eligible for
alert queue will be investigated fourth. Finally, the ef-
fect of scheduling a 31 day month will be examined.

ADDITIONAL PERFORMANCE MEASURES

Two additional performance measures were built to see
how they affected the selection of the internal factor set-
tings. The second performance measure equally weights all
attributes. This performance measure is designed to be a
baseline or "middle of the road" performance measure. The
third performance measure is designed to be the opposite of
the original performance measure used in the experiment.
This performance measure heavily weights morale and lightly

weights training. Both of these performance measures are outlined in Figure IV-4 a and b.

The data points from the experiment were used in this analysis and the performance measure values were recomputed using the two new performance measures. The same graphical and statistical analysis were applied to the two new performance measures used in the experiment. The results indicate that the same internal factor settings are best for all three performance measures. Graphs of the response surfaces for these analyses are included in the back of this chapter as Figures IV-10 through IV-13.

The results of these analyses show that changing performance measures does not change the internal factor settings. These results indicate that the scheduling model internal factors maybe insensitive to changes in the performance measure. If this is true, the internal factor settings are independent of the performance measure and could be put into the model as constants.

INCREASING THE NUMBER OF ACP/SCP CREWS

The original hypothesis was that an increase in the number of SCP qualified crews would increase the performance measure value of the schedule by decreasing the variance in the alert rate of the crew force. Twenty one data points from the original experiment were repeated with the entire crew force ACP/SCP qualified. These data points, their means, and variances are summarized in Table IV-4.

a. Performance Measure Two

MAINTAIN CREDIBLE
DETERRENT CAPABILITY (1.0)

TRAINING (.33) RESOURCE (.33) MORALE
EFFECTIVENESS UTILIZATION (.33)

MPT TIME CLASSROOM ALERT ALERT INTEGRAL FREE
(.50) EFFECTIVENESS RATE VARIANCE ALERTS DAYS
  (.50) (.50) (.50) (.50) (.50)

b. Performance Measure Three

MAINTAIN CREDIBLE
DETERRENT CAPABILITY (1.0)

TRAINING (.20) RESOURCE (.40) MORALE
EFFECTIVENESS UTILIZATION (.40)

MPT TIME CLASSROOM ALERT ALERT INTEGRAL FREE
(.60) EFFECTIVENESS RATE VARIANCE ALERTS DAYS
  (.40) (.50) (.50) (.60) (.40)

Figure IV-4 a and b. Objectives Hierarchy for Performance
Measures Two and Three

The analysis of the data points is included in Appendix B.

The results indicate that the performance measure value

actually decreases or remains the same for an all ACP/SCP

Table IV-4. Results of Analysis of Maximum versus
             Minimum SCP Force

Performance Measure	Mean		Variance (xE-4)	
	Min	Max	Min	Max
1	.921	.918	1.17	.679
2	.855	.843	4.87	3.19
3	.820	.800	10.2	10.4

crew force. The reason for this is a decrease in the inte-
gral alert rate. The integral alert rate decreases because
all crews now perform alerts at three additional launch
control centers. The alert rate and the variance in alert
rate are not changed by increasing the number of SCP crews.
This also indicates that increasing the number of SCP qual-
ified crews can only decrease the performance measure value
of the schedule. One caveat, however, must be placed on
this conclusion. The results were obtained by looking at a
schedule with no leave distribution assumed for the crew
force. The results of this analysis may be different for a
schedule where some leave distribution is assumed. SCP
crews on leave could increase the alert rate of the remain-
ing SCP crews unless there were extra crews to help perform
SCP alerts.

ADDING LEAVE TO SCHEDULE INPUTS

The effect of adding an assumed leave distribution to

the scheduling inputs is discussed in this section. Eleven

data points from the original analysis were re-run with an

assumed leave distribution for the crew force inserted into

the scheduling program. These data points, means, and vari-

ances are presented in Table IV-5. The schedule from the

44th Strategic Missile Wing that was used for verification

and validation was examined and a leave distribution for the

crew force was postulated. The data points used and the

statistical calculations performed are also contained in

Appendix B.

The results of this analysis show a decrease in the per-

formance measure value, but no change in the shape of the

response surface. The decrease in performance measure value

is due to an increase in the variance in alert rates and a

decrease in the MPT rate. Also several crews must be sched-

uled for individual classroom training because of the large

amount of leave taken. The analysis also shows that the

program is capable of functioning with an assumed leave

distribution. The ability of the program to function with a

leave distribution was also demonstrated in the validation

section of this report. This, once again, shows that the

program is capable of building a feasible schedule.

CHANGING THE EA QUEUE PRIORITY RULES

Two different priority rules were tested for the elig-

ible for alert (EA) queue. The first rule was Last In First

Out (LIFO) and the second was First In First Out (FIFO).

Table IV-5. Results of Analysis of Leave vs. Non-Leave
            Schedules

| Performance | Mean | | Variance (xE-4) | |
Measure	Leave	No Leave	Leave	No Leave
1	.892	.924	1.62	.734
2	.829	.859	3.45	2.83
3	.785	.827	5.99	7.53

A LIFO priority rule increases the alert variance for line crews to over 4.0. The reason for this increase is that crews returning to the EA queue from alert have the highest priority for selection to go back on alert. The same crews ar performing a number of back-to-back alerts before any other crews are considered for alert duty. New crews are selected for alert only as crews performing back-to-back alerts complete their maximum number of alerts. This causes a large imbalance in the number of alerts a given crew might perform, driving the variance in alert rate up. This is not a useful strategy, based on the performance measures used in this report because it always reduces the performance measure value of the schedule.

The FIFO priority rule is very similar to the fewest alerts priority rule that was used in the experiment. With the FIFO rule, crews are all cycled through the EA queue before a crew that has previously performed alert is selected. The attribute values recorded for this priority rule are identical to the attribute values seen in the

experiment for the same internal factor settings. This indicates that using the FIFO rule produces the same results, in the scheduling policies examinec as the fewest alerts rule.

THE 31 DAY MONTH

Several program runs were made using a 31 day month. An increase in the alert rate and free days was noted because of the extra day. Adding an extra day to the month constrains the programs ability to find crews for alert, because many crews have already performed their maximum number of alerts by the 31st day. The number of ACP/SCP qualified crews is a particular problem. If there is an inadequate number of ACP/SCP crews the SCPs may not be manned on the 31st day. For a crew force of 39 ACP/SCP qualified crews, however, the program did not have a problem. Once again, this analysis was performed with no leave assumed. The results may be different with a leave distribution.

RESULTS

The internal factors of the scheduling program are robust to changes in the performance measure used to judge the schedule. This means that these factors may not have to be changed for each new performance measure. Adding an assumed leave distribution to the schedule inputs does decrease the performance measure value of the schedule, but the response

surface generated is the same shape.

Increasing the number of ACP/SCP personnel decreases the performance measure value of the schedule because the integral alert rate is decreased. This result may be different if a leave distribution is assumed. If a number of ACP/SCP crews are on leave at the same time there could be a problem manning all SCP launch control centers, forcing the alert rate up for SCP qualified crew members.

One external factor and two optional factors were not addressed in the sensitivity analysis section. It was believed that changing the total number of crews in the crew force would be of little benefit in this analysis. Any decrease in the size of the crew force will increase the alert rate and decrease the performance measure value. Therefore, the best crew force size is the biggest crew force possible. Changing the maximum number of alerts for a crew type would improve the performance measure value of the schedule, but the suggestion is not practical. Flight commander crews and DOT/DOV personnel are responsible for many other duties that preclude an increase in their alert rate. Increasing the maximum number of alerts for line crews should have no effect as they already perform whatever alerts are necessary. Finally, changing the maximum and minimum values of class sizes is not an important consideration. A class size range of between 6 and 15 crews is very large and is typical of class sizes seen at SMWs now.

The results from this effort, recommendations, and
conclusions are presented in the next chapter.
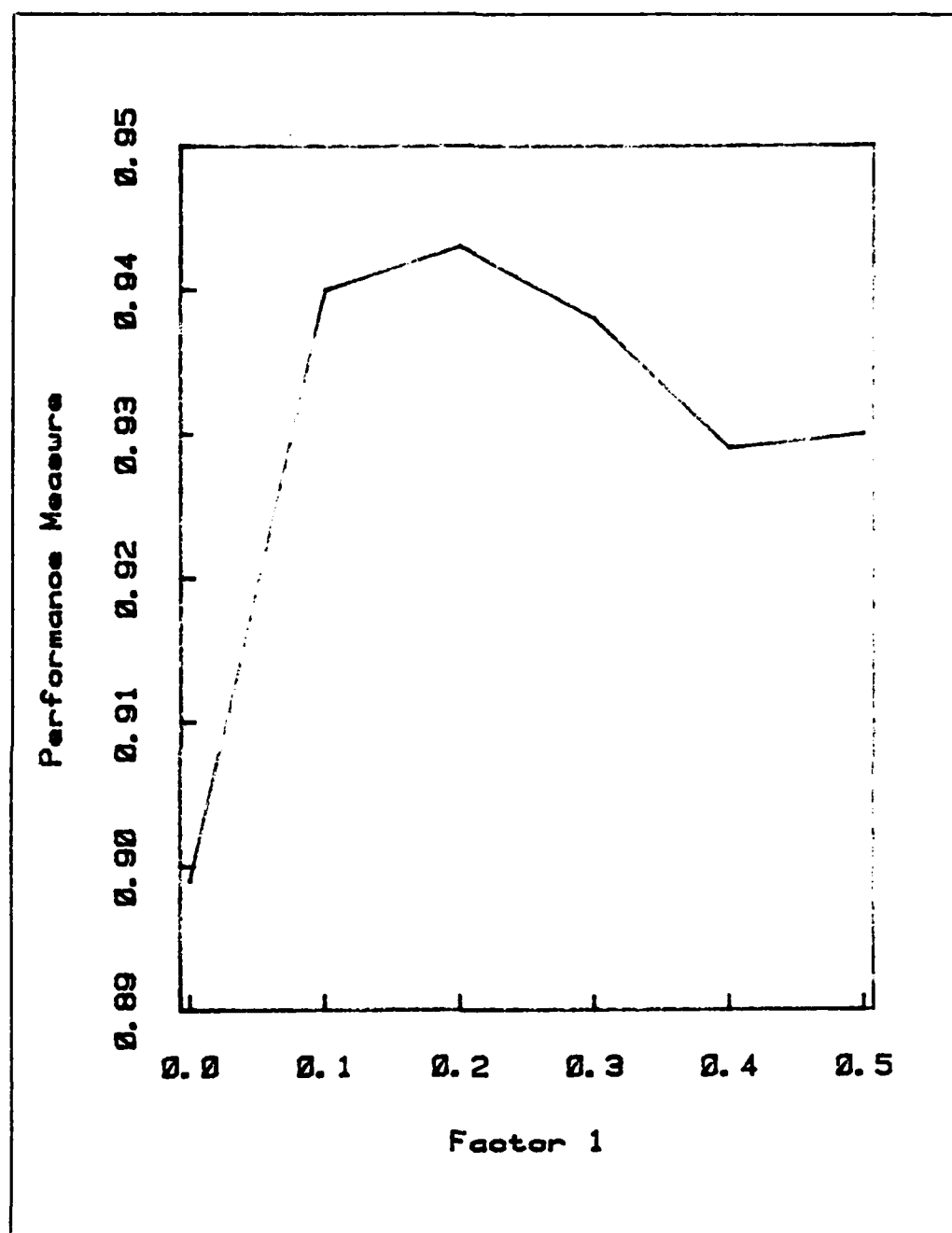
Figure IV-5. Factor One vs. Performance Measure
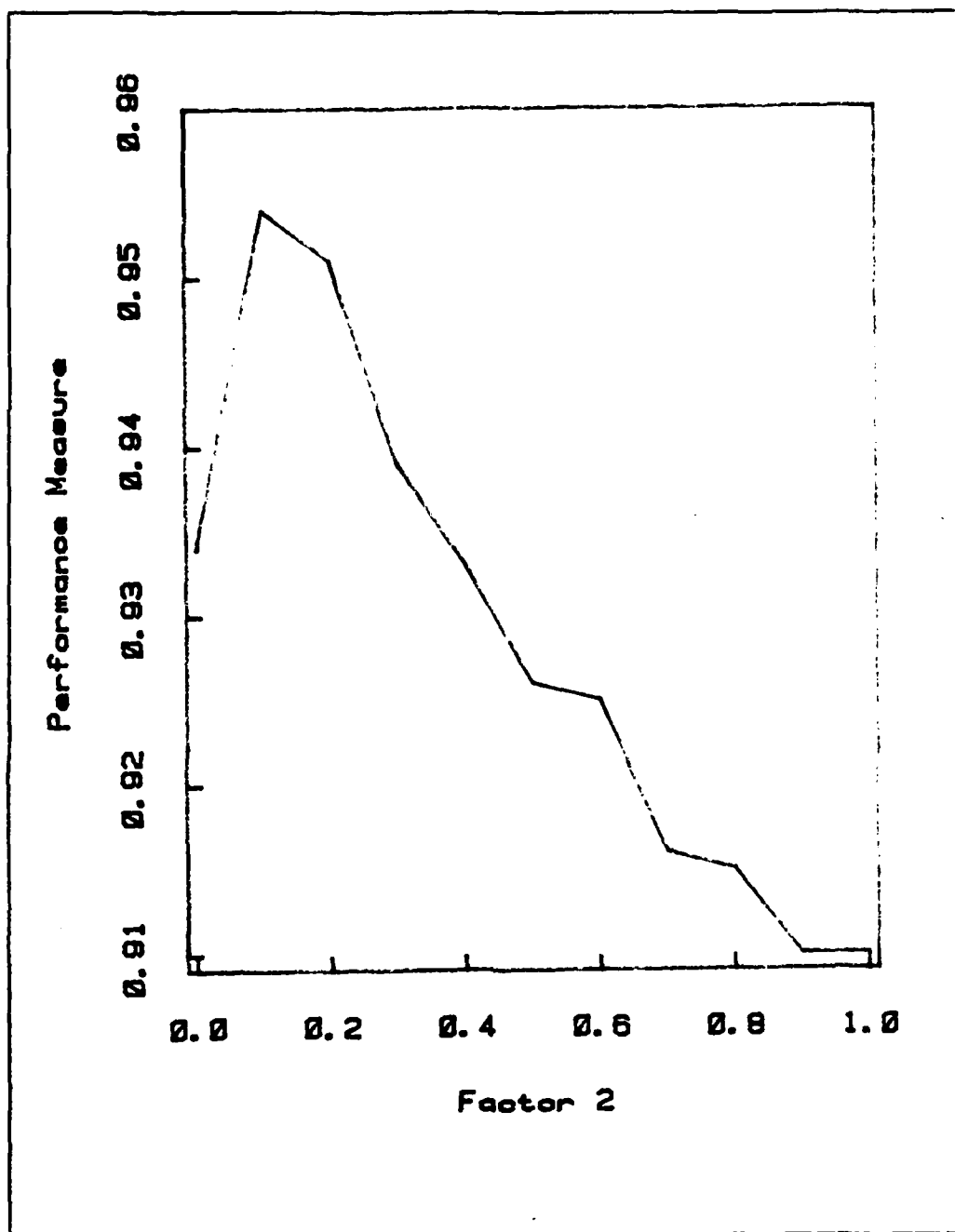
Figure IV-6.   Factor Two vs. Performance Measure
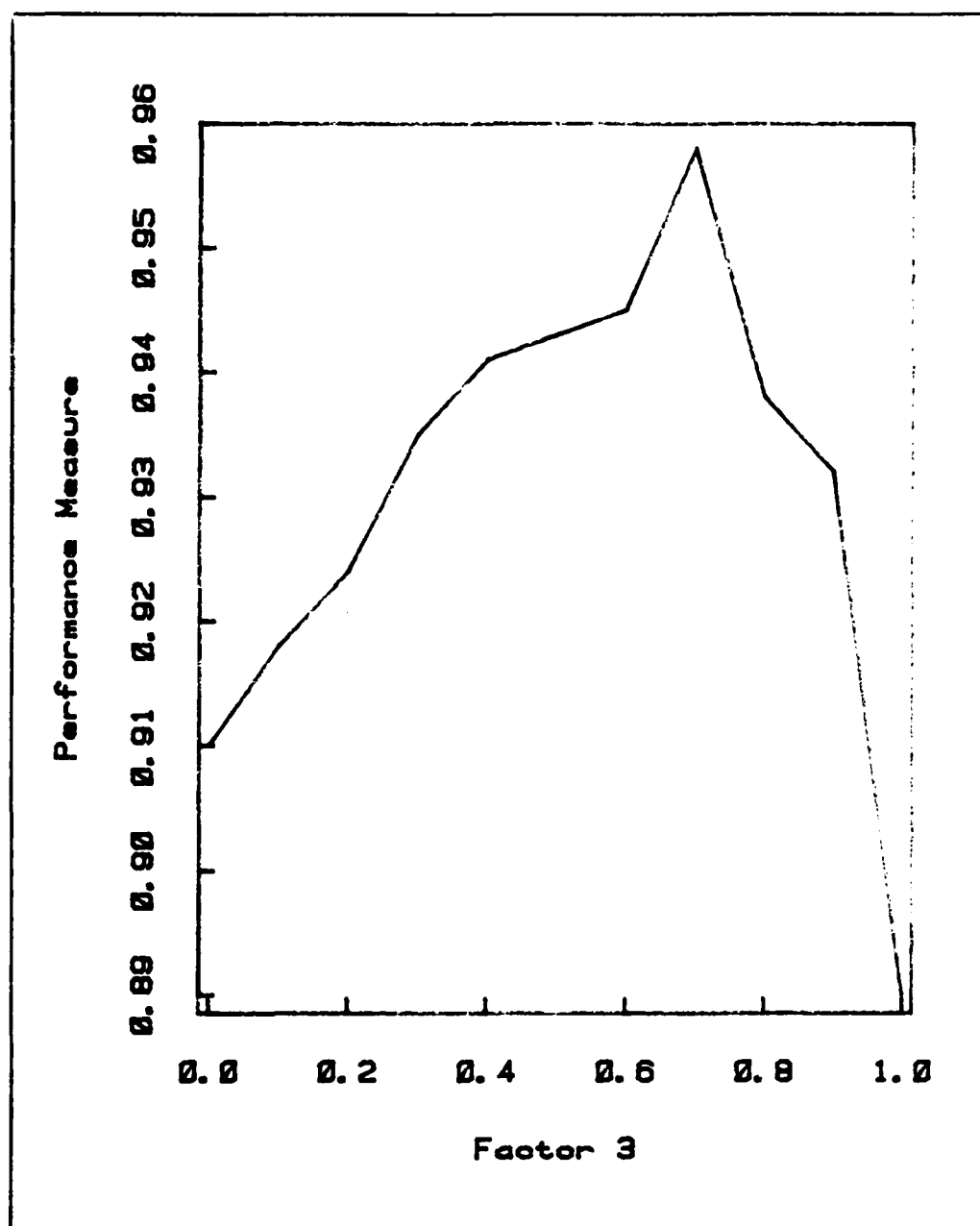
Figure IV-7. Factor Three vs. Performance Measure
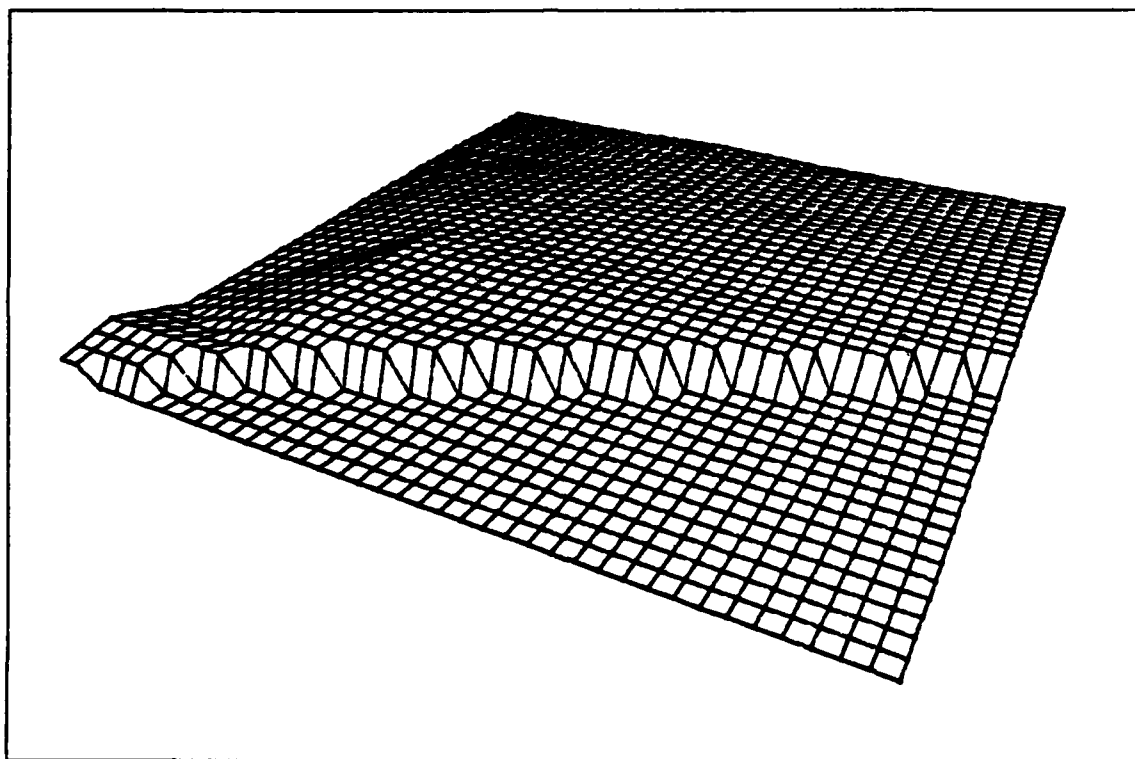
Figure IV-8.   Factors One and Three vs.
Performance Measure One
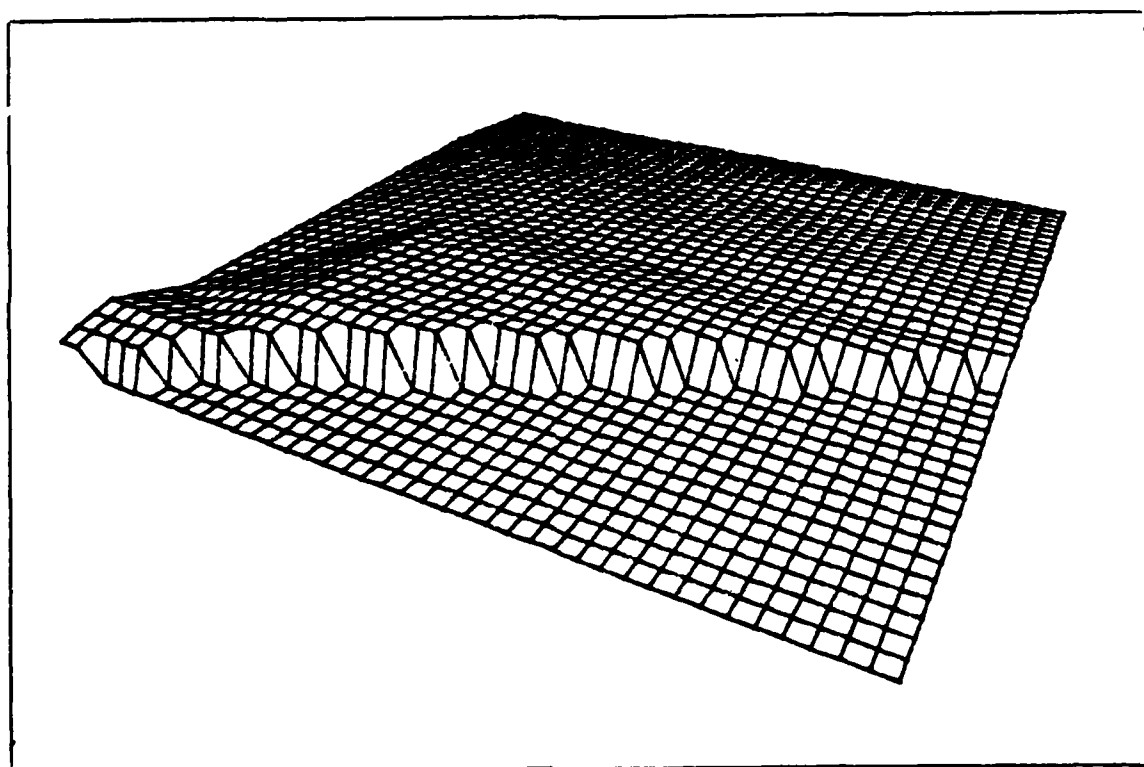
Figure IV-9.   Factors One and Three vs.
               Performance Measure Two
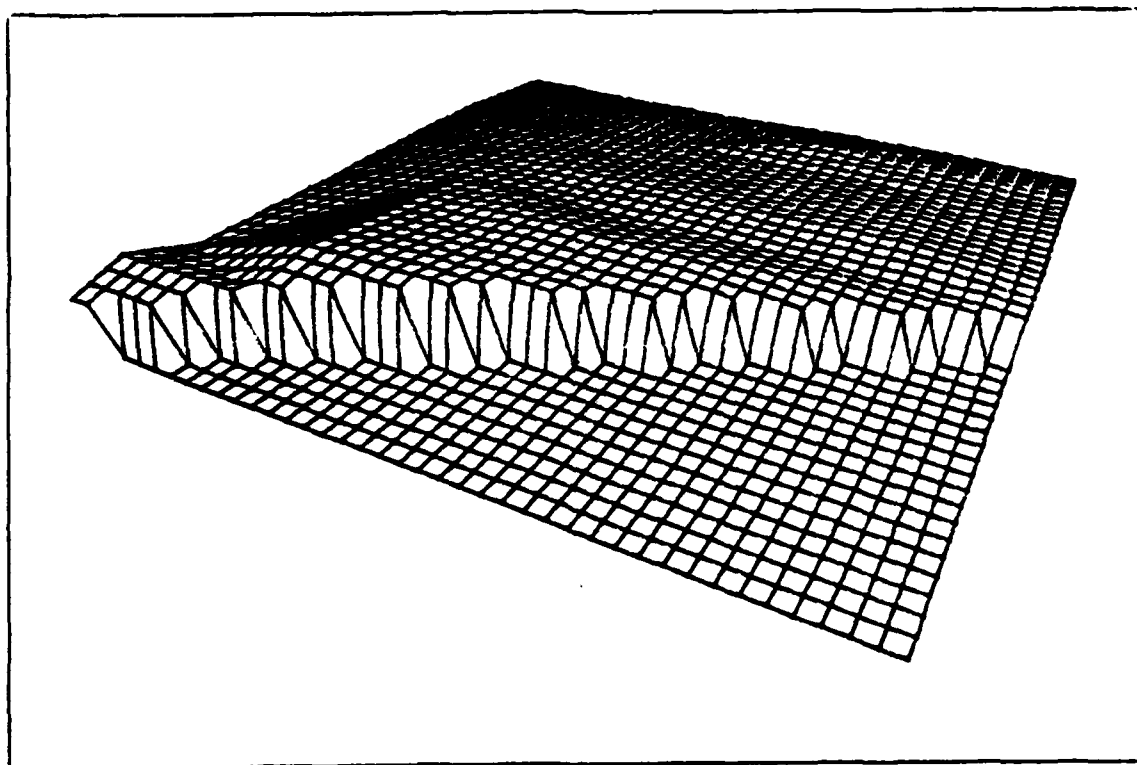
Figure IV-10.   Factors One and Three vs.
                Performance Measure Three
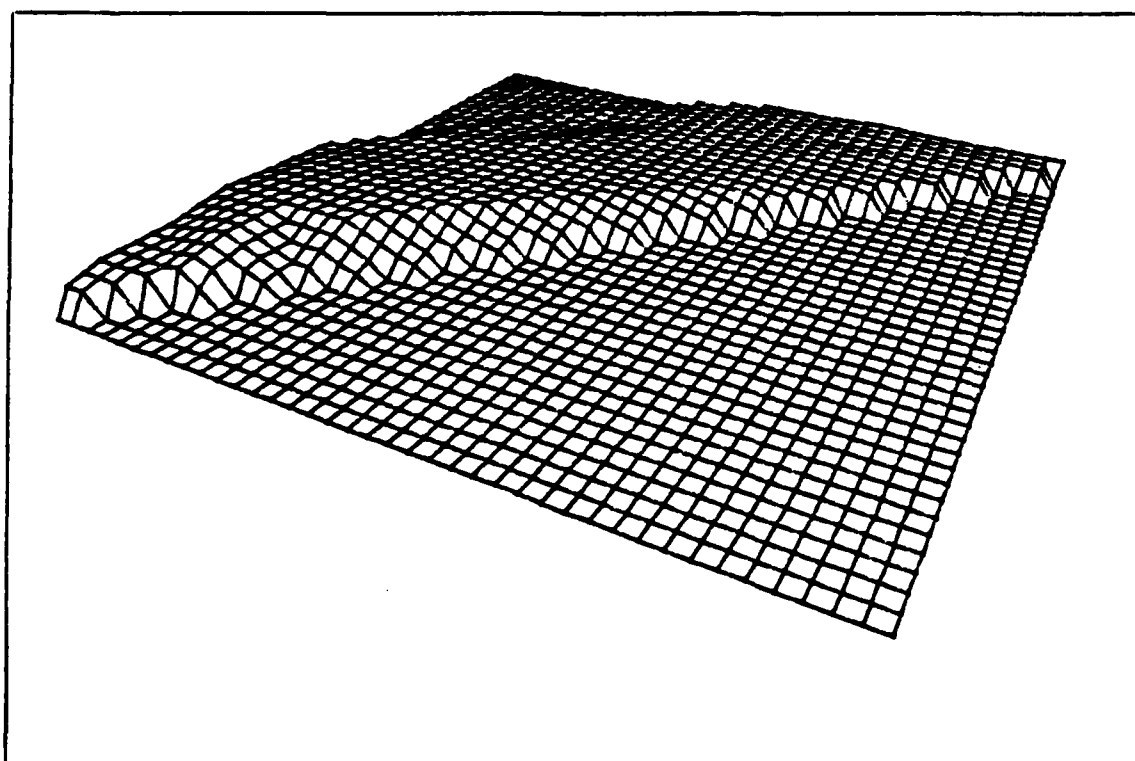
Figure IV-11. Factors Two and Three vs.
                Performance Measure One
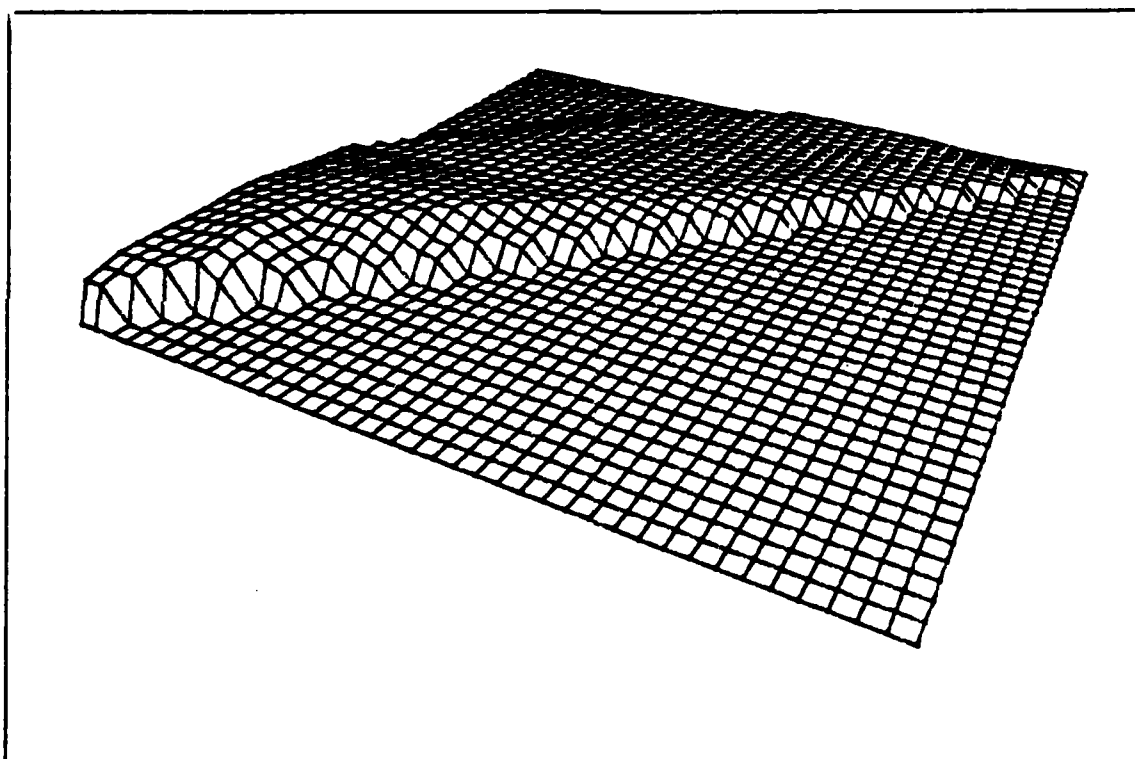
Figure IV-12.   Factors Two and Three vs.
                Performance Measure Two
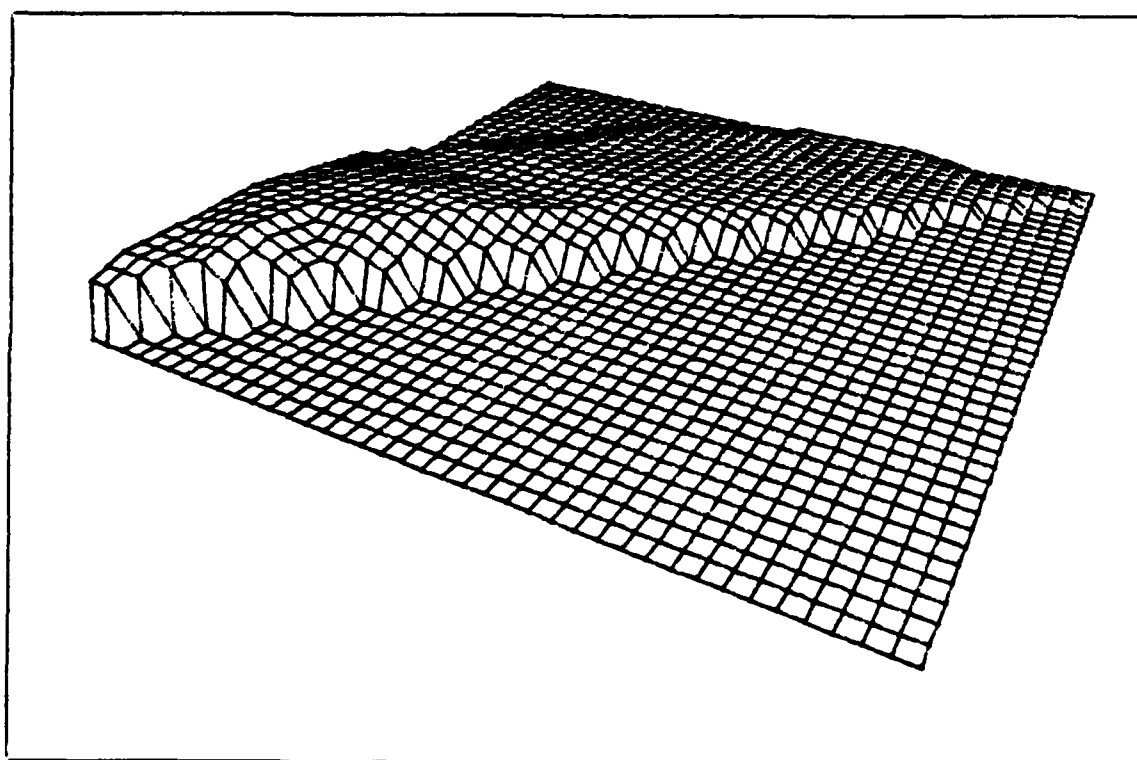
Figure IV-13.   Factors Two and Three vs.
                Performance Measure Three

SUMMARY AND CONCLUSIONS

SUMMARY OF RESEARCH

This research effort was directed toward the development
of a computerized scheduling system for missile combat crews
at a strategic missile wing. The scheduling program takes a
number of inputs from the user and returns a monthly sched-
ule. The user must input several pieces of information
about the crew force including the number of crews, the num-
ber of each type of crew, the home launch control center of
each crew, the number of leave periods the crew has, and the
duration of each leave period. The number of days available
for classroom training to be given must also be input along
with the different type of classroom training. The class
size range for each type of classroom training must be input
also. The number of MPT periods available each day, and the
number of MPT sessions that each crew should receive are
other inputs. The user must also provide a maximum number
of alerts for each crew type. In addition to these inputs,
the user must also set five internal factors. These factors
control the program's method of searching for the next crew
to go on alert. These factors include:

> Factor 1: The percentage of time a DOT/DOV crew is
>
> chosen to perform the next alert at a non-

SCP launch control center (LCC).

Factor 2: The percentage of time a flight commander crew is chosen to perform the next alert at the crew's home LCC.

Factor 3: The percentage of time a crew is chosen to perform alert at their home LCC.

Factor 4: The percentage of time a DOT/DOV crew is selected to perform the next alert at an SCP.

Factor 5: The percentage of time a crew is selected to perform alert at their home LCC that is an SCP.

The output from the scheduling program is an initial monthly operations schedule. The program has scheduled alerts, training, and leave for all crews. The program prints out a series of statistics in addition to the monthly schedule. These statistics include the alerts, integral alerts, leave days, and free days for every crew. Also the mean and variance of alerts, integral alerts, and free days are printed for the line crews.

Some manual manipulation of the schedule is usually necessary, depending on the inputs to the program. This manual reworking of the schedule does not take a significant amount of time.

An example has been presented of how to compare alternative schedules through the use of a performance measure. The method was developed to enable schedulers to produce several schedules, to compare these schedules, and to select

the schedule that best matches the user's criteria of a good
schedule.

RESULTS

An experiment was conducted that compared schedules, and
the results were used to find the best set of internal fac-
tors for the performance measure used. Over 200 data runs
were completed in the experiment and over 50 data points
were gathered. Two other performance measures were devel-
oped, and the data was used to find the best factor settings
for these measures also.

Additionally, 11 data points were gathered with a leave
distribution for the crew force. The results of these runs
were compared to the experimental results. An additional 21
data points were checked with a different percentage of SCP
qualified crew members. These runs were also compared to
the experimental results.

A large amount of data was gathered concerning the
scheduling program. This data was analyzed to better under-
stand how the program works and how to produce the best pos-
sible schedules with the program.

RESEARCH METHODS

The SLAM simulation language was used to build the
scheduling program. For alerts, missile combat crews were
modeled as entities flowing through a network. Other ac-

tivities, such as leave and selection of the next crew for alert were modeled as discrete events. Classroom and MPT training were scheduled following the simulation of the month's alerts. Finally, crew statistics were collected from the schedule.

Crews were scheduled for leave based on the information given to the program by the user. Crews were scheduled for alerts by using search routines that selected the next crew for alert based upon the internal factors, and a priority rule. The priority rule defines which crew should be selected for the next alert. The priority rule used in the experiment was that the crew with the fewest alerts was given highest priority to be selected for alert. Crews were scheduled for classroom training by using a search routine that seeks to minimize the number of training days used within the constraint of a given class size range. Crews were scheduled for available MPT sessions with a similar search routine. Statistics were calculated after the schedule had been completed.

The program was verified by checking each program module output against what was expected. The program was built in modules to facilitate this process. The program modules include: input, initialization, scheduling leave and alerts, scheduling classroom training, scheduling MPT sessions, computing statistics, and program output.

The program was validated by comparing its output to an existing missile combat crew schedule from Ellsworth AFB.

The scheduling program was given the inputs to this schedule and the output closely matched (actually performs better than) the existing schedule.

Worth assessment, a Multi-Criteria Decision Theory technique was used to develop a performance measure for the schedules. This technique uses a six step process for developing a performance measure. The six steps include:

1. List the objectives for the schedule.

2. Develop physical measures for these lower level attributes.

3. Determine the relationship between the lower level attributes and the physical measure.

4. Determine the relationship between the physical measure and the worth of that attribute.

   a. The worth of the attribute is described by a number between zero and one.

   b. The relationship between the maximum and minimum values of the physical measure and the worth score must be linear.

5. Determine the relative weights of the attributes.

6. Adjust these weights as necessary to insure that the decision maker has confidence in the performance measure.

Response surface methodology was used to compare the alternative schedules, and to find the best internal factor settings for the performance measure developed. Response surface methodology allows several factors to be varied to

find the best yield (in this case a performance measure value).

Three different performance measures were compared by testing the means and variances of the data runs to see if the performance measure values differed significantly from one another. The results from changing the number of SCP qualified crews and from assuming a leave distribution were compared in the same manner. Both of these changes were compared to the experimental performance measure values developed first.

CONCLUSIONS

The primary result of this research was the determination that SLAM can be used to develop a feasible missile combat crew scheduling program. This schedule can be produced much faster than by the manual method now used. In addition, the results indicate that the schedule may be better depending on the performance measure used to judge the schedules.

A method was also developed that allows quantitative comparison of scheduling alternatives. The method also can be used to determine the best setting for the internal factors of the model for a given performance measure. The internal factors of the model were found to be robust across several different performance measures. This tends to indicate that the internal factor setting can be adjusted for any given performance measure. The experiment conducted

also shows that the response surfaces of the various performance measures have the same shape.

The introduction of an assumed leave distribution decreases the value of the performance measure observed. An increase in the alert variance and a decrease in the MPT session percentage is the cause of this. A smaller crew force would always have a lower performance measure value because the program produces a schedule that has a near minimum alert rate. Any decrease in the crew force would cause an increase in the alert rate, and a corresponding decrease in the performance measure value.

An all SCP qualified crew force also is not desirable, although the reason for this is less obvious. An all SCP crew force causes a decrease in the integral alert rate causing a lower performance measure value. When all crews are SCP qualified this increases the number of different LCCs where the crew must perform alerts, thus lowering the integral alert rate. Since this analysis was conducted without an assumed leave distribution for the crew force, some additional SCP crews would improve the performance measure value of a schedule, especially if a large number of SCP crews were on leave at one time. In this case, additional SCP crews would lower the alert rate and the variance in alert rate and would offset the decrease in the integral alert rate.

One final lesson learned from this research effort is the need for a data base for this scheduling model. A data

base could hold all the crew information that must currently
be input into the scheduler every time it is run. The data
base could also hold additional information on arrival and
departure of personnel, periodic appointments, and other
constraints on the scheduling program. This data base could
allow rapid changes to the schedule. Without a data base
that a scheduler can use to call up information and make
decisions, the program is of limited use in making day-to-
day schedule changes. With a data base, a scheduler could
call up the needed information, decide how to change the
schedule, and input the change into the scheduling program
rapidly. Such a system is a Decision Oriented Scheduling
System, and would be of great value to missile operations
and missile maintenance schedulers.

USES FOR THE SCHEDULING PROGRAM

As written, the scheduling program gives wing schedulers
the ability to quickly produce a high quality, initial
monthly schedule. The program also can produce several al-
ternative schedules that can be compared in the manner de-
scribed in Chapter IV of this report. The program cannot be
used for making daily schedule changes, however, without the
addition of a data base.

The program also can be used to forecast crew require-
ments. The future requirements can be input into the pro-
gram and various force structures can be examined. The best
force structure can be chosen and employed. This method

would work even better if the program was provided with the data bases previously discussed.

RECOMMENDATIONS FOR FURTHER RESEARCH

This research effort can be thought of as the first part of a Strategic Missile Wing Decision Oriented Scheduling System (DOSS). The complete DOSS would include a missile operations scheduler and data base; a missile maintenance scheduler and data base; a missile status data base; an interface between the data bases and the scheduler, and an interface between the DOSS and the users. This DOSS would decrease the amount of time currently spent scheduling missile operations and maintenance requirements. Information about the various components of the missile wing would be readily available allowing decisions to be made with the latest and most complete information.

The scheduling program also may require a different method of developing a performance measure. Other MCDT techniques are available that would provide a more detailed performance measure for the decision maker. These techniques, however, are time consuming and require a great deal of research. Value functions would have to be elicited from wing decision makers as a starting point in the construction of more accurate performance measures.

In addition, more sensitivity analysis is needed for the scheduling program itself. The effects of changing leave distributions, changing the MPT requirements for each crew,

and changing to different performance measures should be addressed. A method of making rapid daily changes to the schedule also must be developed. Finally, a method of using the schedule for forecasting has been mentioned, but the method has yet to be developed. The scheduling program could be a valuable tool for forecasting because of the ability to make alternative schedules rapidly.

This research effort has taken an initial step into the development of a DOSS for strategic missile wings. A considerable amount of research lies ahead before the idea can become a reality. The results of the research are, however, quite useful now. The program can be applied to solve several recurring management scheduling problems in strategic missile wings.

Bibliography

1. Ackoff, Russell L. *Progress in Operations Research.*
New York: John Wiley and Sons, Inc., 1961.

2. Angell, D. R. "An Analysis of Missile Crew Scheduling
in SAC." Unpublished research report. Air War College,
Maxwell AFB, Alabama, April 1977. (AD-B017-180L).

3. Arthur, J. L. and A. Ravindran. "A Multiple Objective
Nurse Scheduling Model," *AIIE Transactions,* 13 (1): 55-60
(March 1981).

4. Barnidge, L. and B. H. Cioli. *The Wing Level
Scheduling Process: Systems Approach.* MS thesis.
Wright-Patterson AFB, Ohio: Air Force Institute of
Technology, September 1978. (AD-A060-449).

5. Bartholdi, J. J. "A Guaranteed Round-Off Algorithm for
Cyclic Scheduling and Set Covering," *Operations Research,*
29 (3): 501-510 (May-June 1981).

6. Berman, M. B. "Improving SAC Aircrew and Aircraft
Scheduling to Increase Resource Effectiveness." RAND
Corporation report. Santa Monica, California, July 1974.
(AD-787-206).

7. -----. "Scheduling Aircrews and Aircraft: Problems of
Resource Allocation." RAND Corporation report. Santa
Monica, California, January 1975. (AD-011-613).

8. Bush, M. D. "Automated Missile Operations Management
System." Unpublished report. 351st Strategic Missile Wing,
Whiteman AFB, Missouri, 8 June 1976.

9. Chiplunker, A. V., *et al*. "Optimization of Refuse
Collection Systems." 1203-1210. *American Society of Civil
Engineering Proceedings* 107 (EE6, #16731), December 1981.

10. Crandall, K. C. "Analysis of Schedule Simulations."
387-394. *American Society of Civil Engineering Proceedings*
103 (CO3, #13202), September 1977.

11. Crandall, K. C. and J. C. Woolery. "Schedule
Development Under Stochastic Scheduling." 321-329.
American Society of Civil Engineering Proceedings 108 (CO2,
#17120), June 1982.

12. Cunto, E. "Scheduling Boats to Sample Oil Wells in
Lake Maracaibo," *Operations Research,* 26 (1): 183-196
(January-February 1978).

13. DeWispelare, Aaron. Class notes from ST 6.31, Effectiveness/Trade-Off Studies of Weapon Systems. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1983.

14. Dutton, J. M. and W. H. Starbuck. *Computer Simulation of Human Behavior.* New York: John Wiley and Sons, Inc., 1971.

15. Fallon, R. "Rule-Based Modeling as an Analysis Tool: Implications for Resource Allocation within Strategic Air Command." RAND Corporation report. Santa Monica, California, April 1980. (AD-A085-394).

16. Hershaur, J. C. and R. J. Ebert. "Search and Simulation Selection of a Job Sequencing Rule," *Management Science,* 21 (7): 833-835 (March 1975).

17. Hillier, F. S. and G. J. Lieberman. *Introduction to Operations Research.* San Francisco: Holden-Day Inc, 1980.

18. Hosios, A. J. and J. M. Rousseau. "A Heuristic Scheduling Algorithm," *Journal of the Operations Research Society,* 31: 749-753.

19. Mageath, J. D. "Successful Personnel Scheduling Problems," *Interfaces,* 8 (2): 55-60.

20. Martel, C. "Preemptive Scheduling with Release Times, Deadlines, and Due Times," *Journal of the Association for Computing Machinery,* 27 (3): 812-829 (July 1982).

21. McGough, E. H. "Scheduling: Effective Methods and Techniques." 75-84. *American Society of Civil Engineering Proceedings* 108 (CO1, #16900), March 1982.

22. Moore, J. M. and R. D. Whitmoore. *A SAC B-52 Aircrew Scheduling Model Using ICAMS IDEF Methodology.* MS thesis. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, September 1982. (AD-A060-449).

23. Mortenson, R. E. "Maintenance Planning and Scheduling Using Network Simulations," *Winter Simulation Conference Report,* Vol 1. 333-340. Atlanta, Georgia, December 1981.

24. Myers, R. H. *Response Surface Methodology.* New York: John Wiley and Sons, Inc., 1976.

25. New, C. C. "Job Shop Scheduling: Is Manual Application of Dispatching Rules Feasible?," *Operations Research Quarterly,* 261 (1): 33-45 (March 1975).

26.   Pease, M. C. "The Schedulers of ACS.1." Stanford
Research Institute report for the Department of the Navy.
Arlington, Virginia, November 8, 1977.

27.   Pritsker, A. Alan B. and Claude Dennis Pegden.
Introduction to SLAM and Simulation. New York: John Wiley
and Sons, Inc., 1979.

28.   "Scheduling Rules Study LGM30F SAC." Startegic Air
Command unpublished report, September 1975.

29.   Schoderbeck, C. G., et al. Management Systems
Conceptual Considerations. Dallas: Business Publications,
Inc., 1980.

30.   Shannon, R. E. Simulation the Art and Science. New
Jersey: Prentice Hall, Inc., 1975.

31.   Warner, D. M. "Scheduling Nursing Personnel According
to Nursing Preference: A Mathematical Programming
Approach," Operations Research, 24 (5): 842-856
(September-October 1976).

32.   Yamayee, Z. A. "Maintenance Scheduling, Description,
Literature, and Interface with Overall Operations," IEEE
Transactions on Power Apparatus and Systems, PAS-101 (8):
2770-2779 (August 1982).

APPENDIX A

USER'S GUIDE

INTRODUCTION

This user's guide is presented in three parts. Part I
provides information to the user who is familiar with the
scheduling of missile combat crews (MCC) but not proficient
in computer programming. Part II, on the other hand,
provides a detailed explanation of the programs used to
input crew data and build the monthly schedule, and assumes
a working knowledge of both FORTRAN and SLAM programming
procedures. Finally, Part III contains the program and
variable listings.

PART I

GETTING STARTED

Prior to using the crew scheduling program, crew infor-
mation must be assembled into a form useable by the program.
Crews numbers should be assigned in sequence from 1 to a
maximum of 100. In addition, the program requires flights
1, 6, and 11 to be the squadron command posts (SCP). For
maximum efficiency, any recrewing actions should be complete
on the first day of the month. Also, reserve or spare crew
members should be crewed together for scheduling purposes,
when possible.

The scheduling program provides for as many as three
leave periods for each crew. Although referred to as leave
by the program, these periods can be used to reflect re-
quested days off other than leave or crew non-availability
days. These non-availability days can be used by a sched-
uler to represent crew training, such as upgrade or SCP
training, temporary duty assignments, or duty not including
alert (DNIA) restrictions.

The training information required for the program in-
cludes the number of different types of classroom training
required, the dates each will be offered, and the number of
Missile Procedure Trainer (MPT) sessions per crew. A maxi-
mum of 15 dates of offering for each type classroom train-
ing may be specified. In addition, the maximum and minimum
class size must be specified. The total number of MPT peri-

ods per day available for training must also be known before
using the program.

Finally, an executable file of the program SLMWRT must
be available. This program provides an interactive environ-
ment for the scheduler to enter the monthly requirements for
the crew scheduling program.

ENTERING THE DATA

The program SLMWRT provides a user friendly method of
entering the information required to build the monthly
schedule. Once the information discussed above is avail-
able, simply run the program and answer the questions. In
addition to the instructions provided in this guide, a num-
ber of checks have been provided in the program to decrease
the chances of entering inconsistent information.

The following is an example of an interactive session
using the SLMWRT program. Program prompts are shown in
upper case and a typical input is shown in lower case.

ENTER DATE (EG, 10/12/83)
02/15/84

ENTER PRIORITY FOR CREW FILE (EG, LVF(5))
lvf(5)

ENTER TIE-BREAKER PRIORITY
lvf(4)

These two entries shows the normal priority of se-
lecting crews from the eligible for alert (EA) queue. The
first entry means the crew with the fewest number of alerts
(attribute 5) will be selected first. The tie-breaker rule
specifies that if two crews have the same number of alerts
then select the crew with the smallest value for crew type
(attribute 4) first. Other possible priorities are low
value first, LVF, or high value first, HVF, on any attri-
bute, last in first out, LIFO, and first in first out, FIFO.

ENTER NUMBER OF DAYS IN THE MONTH
30

These questions provide the information necessary to
construct the model used in simulating a month of alert
duty. The next questions are used to elicit information
concerning each crew and the monthly training data.

ENTER NUMBER OF CREWS
90

Enter the total number of crews, including line,
flight commander, DOV/DOT, and any crews formed from spare
or reserve crew members. The program is designed to handle
up to 100 crews.

FIRST FIFTEEN ENTRIES SENT ON ALERT FOR DAY 1

ENTRY NUMBER 1

ENTER CREW NUMBER, FLIGHT, SCP QUAL, AND TYPE

   5 1 1 0

After printing the reminder that the first fifteen en-
tries are immediately sent on alert, the entry number is
printed and information for each crew is requested. The
above entry indicates crew number 5 is assigned to flight 1
and is an SCP qualified line crew. Acceptable values for
the requested parameters are as follows:

Crew number	1-100
Flight	1-17
SCP Qualification:	
Qualified	1
Not Qualified	0
Crew Type:	
Line	0
Flt CC	1
DOV/DOT	2

Data entered outside of these ranges will result in a
warning message followed by another request for the data in
question.

HOW MANY LEAVE PERIODS REQUESTED (0-3)

   1

A-5

If '0' is entered the program will move to the next crew.

```
        ENTER START AND END LEAVE DATES, PERIOD 1
          10 15


        ENTER NUMBER OF ACTUAL LEAVE DAYS FOR CREW 5
          6
```

The above entries indicate crew five has requested leave from the tenth to the fifteenth of the month. If the actual number of leave days was less than six, this would indicate a non-leave unavailability of the crew. If more than one leave is requested the start and end leave date requests would appear the appropriate number of times. The actual number of leave days request is not printed until all leave dates have been entered.

If the end leave date is earlier than the start leave date a warning will be printed followed by a re-entry request for the information. One day requests are input by entering the day twice as both the start and end leave date. In addition, for leaves involving more than one month, only the current month dates are entered. After the leave data is entered the program will proceed to the next crew entry.

Once all crew information has been input, the program will request training information.

ENTER THE NUMBER OF CLASSROOM TYPE TRAINING (1,2,3)

  2

ENTER TRAINING CODE FOR TYPE 1

  33

ENTER MAXIMUM AND MINIMUM CLASS CLASS SIZE

  15 6

ENTER TRAINING DAY FOR TYPE 1

  7

These entries illustrate a normal number of required classroom training types. For example, emergency war order (EWO) and codes training are taught on the same day and weapon system training is taught by itself. The requested training code is a 2-digit numeric code to identify the assigned training day on the schedule. Because of prior use, numbers 1-15 (LCC identifiers), 66 (MPT identifier), 77 (leave identifier), and 99 (alert return day identifier) should not be used. The maximum and minimum class size is self-explanatory, however, the maximum class size allowed by the program is 25 crews. The training day request will be repeated 15 times to allow up to 15 possible days for conducting the training. If less than 15 possible training days are desired, enter '0' to the additional program

requests.

Following the 15 requests for training dates, the
program will ask the same questions for the next classroom
type training.

ENTER NUMBER OF MPT PERIODS PER CREW (1 OR 2)

2

If the number of MPT periods per crew is '1', the next
question will not be presented.

ENTER MINIMUM TIME BETWEEN MPT PERIODS

5

ENTER NUMBER OF MPT PERIODS PER DAY

DAY 1

6

.

.

.

DAY 30

0

The minimum time between MPT sessions is used to sep-
arate MPT training when more than one session is scheduled.
The number of MPT periods for each day should reflect the
total number of MPT periods available for training. Crews

will be assigned for MPT training up to the maximum number of periods available on that day. The actual times for each session will have to be manually determined by the scheduler.

ENTER MAXIMUM NUMBER OF ALERTS FOR:
LINE CREWS, FLIGHT COMMANDERS, AND DOV/DOT
8 4 2

Simply enter the maximum number of alerts for each crew type. Although the response listed above is the normal setting, in times of restricted crew availability higher numbers may be desirable.

ENTER PERCENTAGES FOR ALERT ASSIGNMENTS:
FOR NON-SCP ALERTS -- DOV/DOT, FLT CC, INTEGRAL
.17 .57 .97

FOR SCP ALERTS -- DOV/DOT, INTEGRAL
.3 .8

These percentages are used to search for a specific crew for alert based on the comparison of a random number between zero and one and these percentages. The response indicates that on the average a DOV/DOT crew will be sent to a non-SCP 17 percent of the time, a flight commander will be sent on alert 40 percent (.57-.17) of the time, and a crew

will be sent to their home LCC 40 percent (.97-.57) of the time. If the search for a particular crew is unsuccessful, another crew will be found by first checking available line crews based on the EA queue priority discussed earlier. If this search is unsuccessful than the first available crew of any type will be assigned. The percentages specified for SCP alerts provide the same type of searches, however, only SCP qualified crews are considered. These setting also help distribute DOV/DOT and flight commander alerts throughout the month. Experimentation has shown the following settings provide the best results: for non-SCP alerts, .1 .2 1. or .2 0. 1. for DOV/DOT, flight commander, and integral per-centages, respectively. For SCP alerts the settings are: .2 1. or .1 1. for DOV/DOT and integral percentages, respec-tively.

ENTER THE 2 RANDOM NUMBER STREAMS (1-10)
1 2

The program allows specification of the number stream used to seed the random numbers used in the non-SCP and SCP alert selection process. Ten streams are available and any combination may be used. By running the scheduling program with different streams slightly different schedules will result.

This completes the entry of data necessary for the crew scheduling program. A file named SLAMIN is created by the

program and contains the data entered by the scheduler.
This file should now be saved for use with the crew
scheduling program.

RUNNING THE PROGRAM

The actual procedures for running the crew scheduling
program are both machine and installation dependent and will
not be discussed here. However, the following general guid-
ance is provided to assist the scheduler in producing a high
quality schedule.

Once the data file has been created a number of program
runs should be made with different seeds. The scheduler can
then compare the schedules and choose the the most appropri-
ate. In addition, alternate schedules can be produced by
varying training dates, MPTs and leave requests.

If the program consistently provides a schedule with a
number of alerts unfilled, the maximum alert parameters
should be changed. Once these changes are made, additional
program runs can be made to find a feasible schedule.

PROGRAM EXPLANATION: SLMWRT

As discussed in Part I of this guide, the SLMWRT program is designed to provide a user friendly interface to the SLAM crew scheduling program. It is interactive and consists of only a main program. The program provides two key types of information to the crew scheduling program. First, the program actually writes the SLAM control cards to the file SLAMIN. Immediately following these control cards, the crew and training data is written to the same file.

The SLAM control cards are used to provide information concerning project description, number of files used, number of attributes per transaction, number of concurrent transactions, queue priorities and initial status. In addition, the control cards describe the network used to simulate the crew alert system. The network diagram for this system is contained in Figure A-1 and the SLAM control cards are listed in Part III of this guide.

The network consists of 15 identical queueing systems. These queues and their accompanying service activities represent the 15 launch control centers (LCC) of a three-squadron wing. Entry to the system occurs at each labeled queue node and a transaction representing the crew is then sent to the service activity to simulate a 24-hr alert. At alert completion the crew transaction enters the event node to determine the completing crew's disposition and triggers
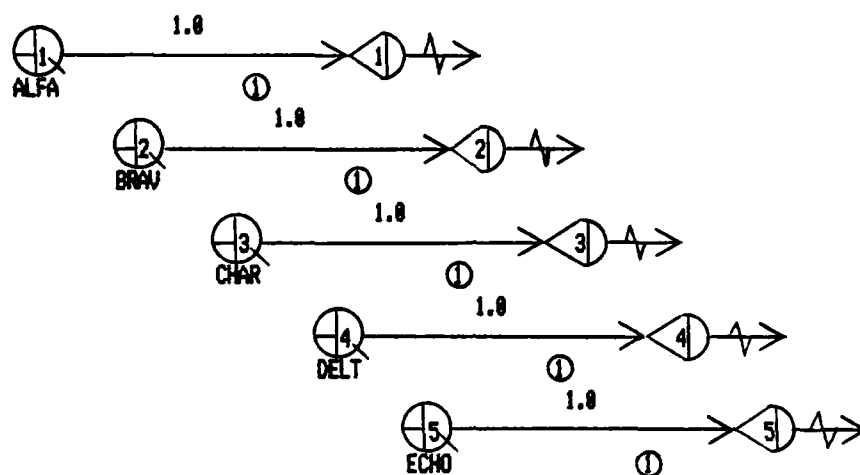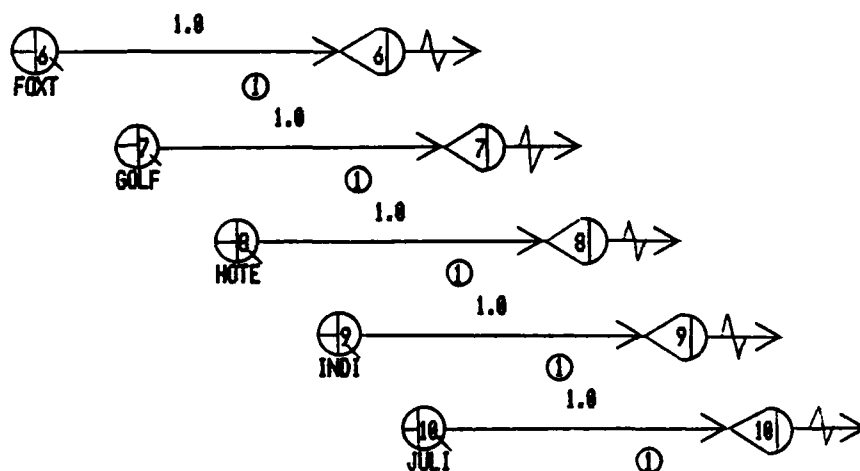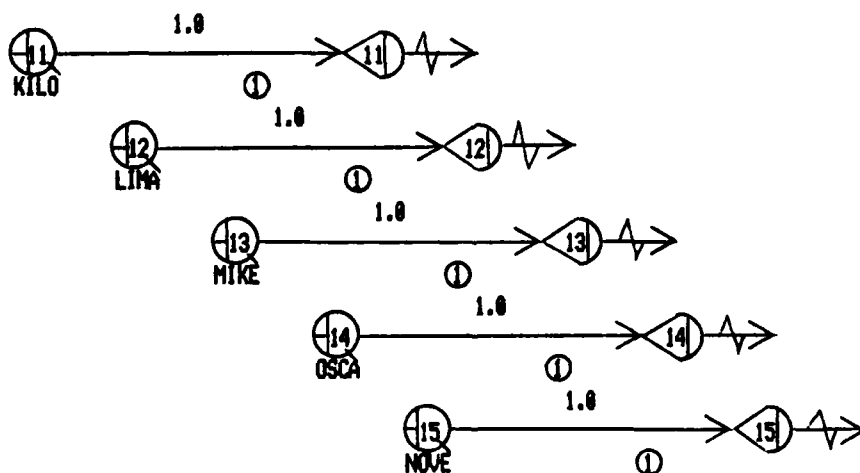
Figure A-1. SLAM Alert Network

a search for the next day's alert crew. These actions occur
when the event node transfers control to the event subrou-
tine of the crew scheduling program. The terminate node is
then used to destroy the completed crew transaction.

The SLMWRT program allows the user to specify the date
of the run for identification, the crew file queueing disci-
pline, and the number of days in the month. Using this in-
formation the SLAM control cards, including the network
cards, are written to the input file. The network is writ-
ten using a simple loop structure which stores the queue
labels in the character array, QUEUE.

The remainder of the program is used to gather informa-
tion on the individual crews and training requirements. The
format consists of a straight forward question and answer
session followed by writing the information to the input
file. To decrease problems associated with incorrect input
data, the SLMWRT program checks most information for cor-
rectness and consistentcy. For incorrect information, a
warning is printed and the user is allowed to reenter the
information. A complete listing of this program is included
in Part III of this guide.

PROGRAM EXPLANATION: CREW SCHEDULING PROGRAM

The crew scheduling program uses a combined network and
discrete event approach to the problem of scheduling crews
for alert. The FORTRAN program consisting of a main program
and ten subroutines provides the logic for the interaction

of the network and discrete event portions of the model. This program will be described below.

Program MAIN

The main program is the standard version used by SLAM to dimension the SLAM arrays and establish input and output device numbers. From the main program the SLAM subroutine is called to control the execution of the simulation.

Subroutine INTLC

The INTLC subroutine is used to initialize user variables and read crew, training, and other scheduling information from the input file created by the SLMWRT program. Named common blocks are used to transfer this information to the other subroutines in the program. Once the INTLC subroutine is complete, the crew scheduling function begins.

An important array initialized in this subroutine is the array DUMMY. This array will be used whenever it is necessary to place an event on the calendar that does not pertain to a specific crew. An array such as DUMMY is needed as a place holder in the call statement of some SLAM subroutines. Specific uses of array DUMMY will be pointed out when they occur.

After initializing user variables and arrays, subroutine INTLC begins to read the crew information from the input file. This identifying crew information is stored as SLAM

```
                        SLAM ATTRIBUTES


    ATTRIBUTE      DESCRIPTION


        1          Crew Number -- must begin at 1 and continue
                   sequentially to a maximum of 100

        2          Flight Designator -- 1 thru 15, 16=DOT, 17=DOV

        3          SCP Designator -- 0=Not qualified, 1=Qualified

        4          Crew Type -- 0=Line, 1=Flight CC, 2=DOV/DOT

        5          Number of Alerts -- Total number of alerts in
                   a month

        6          Spare

        7          LV1 Starting Date

        8          LV1 Ending Date

        9          LV2 Starting Date

       10          LV2 Ending Date

       11          LV3 Starting Date

       12          LV3 Ending Date
```

Figure A-2.   Crew Attributes

attributes (Figure A-2). The total number of crews is read,

followed by the first loop used to enter crew information.

This loop reads the first fifteen crews and sequentially

places one crew in each of the fifteen queues used to simu-

late alerts. These are the crews performing alert on the

first day of the month. In addition, flight assignment data

is copied from attribute 2 to build the INTFLT array and the

actual number of leave days requested by the crew is loaded

into the LVDAYS array. Both of these arrays are indexed by crew number. The INTFLT array will be used in the statistics gathering subroutine to differentiate between line crews (coded with numerical flight assignment, 1-15), flight commanders (coded with the negative of flight assignment, -1-(-15)), and DOV/DOT crews (coded as 16 and 17). The LVDAYS array is also used in the statistics subroutine to account for actual leave days as free days.

The second loop used to enter crew information reads the data for all crews not performing alert on the first day of the month. In addition to building the INTFLT and LVDAYS arrays as discussed above, this loop also checks for any crew with leave beginning on the first day of the month. Crews with leave beginning on the first of the month are placed on the event calendar with a return event scheduled upon completion of their leave. The crew scheduling array, ISCHED, is also updated with the leave information. All other crews are placed in the eligible for alert (EA) queue, FILE 16. A listing of all files used by the program is in Figure A-3.

Following the individual crew information, the crew training data is read. Possible training dates for each type classroom training is placed in array ITRNDY(i,j), where i refers to the type training and j refers to the date index. The numeric code for the training is placed in array MRKDAY and the maximum and minimum allowed class size is stored in the arrays MAXCLS and MINCLS. The number of MPT

```
                        SLAM FILES


   FILE    DESCRIPTION


   1-15    Alert Files -- Represents alert LCCs

    16     Crew File -- Eligible for alert (EA) queue

    17     Inactive Crew File -- contains line crews who have
           completed their maximum number of alerts

    18     Inactive Crew File -- contains DOV/DOT crews who
           have completed their maximum number of alerts

    19     Inactive Crew File -- contains flight commander
           crews who have completed their maximum number of
           alerts

    20     Spare
```

Figure A-3.   SLAM File Description.

sessions requested per crew is read into variable NUMMPT and

the number of MPT training periods available each day of the

month is stored in array MPTDAY. In addition, if more than

one MPT per month is requested the variable for specifying

the minimum number of days between MPT training (MTBMPT) is

read.

Finally, general information needed for the scheduling

program is read from the input file. This includes the

maximum number of alerts allowed for each crew type (MAXLIN,

MAXSHP, MAXFCC), the internal percentages used in deter-

mining selection of crews for alert (SHPPCT, FCCPCT, FLTPCT,

SHPCTS, FTPCTS), and the SLAM random number seed streams

(ISTRM1, ISTRM2).

The INTLC subroutine also establishes the time for checking crew leave requests. This is done by placing an event on the calendar to trigger a call to the leave subroutine. Since this event involves no specific crew, the array DUMMY is used in the call statement.

Subroutine EVENT(IFN)

This subroutine is used to direct program flow in the discrete event portion of the simulation. Arriving transactions carry a code, IFN, used to control their disposition. Transactions can enter this subroutine from the network portion of the model through the EVENT nodes or from the event calendar. Normally these transactions represent crews processing through the model. Dummy transactions, however, are also used to trigger other activities.

As each transaction arrives, the current simulation day (IDAY) and next day (NXTDAY) are recorded. In addition, the variable IFILE is set equal to IFN for transfer to other subroutines. IFILE identifies the queue (LCC) where an alert was just completed for crew transactions. Next the transaction is checked to determine its disposition.

An IFN of 25 indicates a crew transaction has arrived from the event calendar. This indicates the crew has either just returned from leave status or has just completed crew rest requirements. In either case, the crew is placed back in the EA queue and no further subroutine action is required.

An IFN equal to 30 indicates a non-crew transaction from the event calendar. This code is used to schedule the daily leave request check prior to scheduling the next day's alerts. Following the leave check control is returned to the SLAM processor.

The last special case which must be considered is the arrival of a dummy crew transaction. This transaction is necessary to initiate a search for the next alert crew when the previous day's alert could not be filled for some specific LCC. In this case IFILE identifies the LCC and a crew number of zero identifies the dummy transaction. When the dummy transaction arrives it is immediately transferred to the computed GOTO for routing to the appropriate next alert subroutine.

All other transactions represent crews completing an alert. These transactions are input from the EVENT nodes of the network and identify the LCC where the alert was performed by IFILE. First, the crew scheduling array, ISCHED, is updated with the completed alert location and day of return code. In addition, the number of alerts is increased by adding one to attribute 5.

Next the crew is checked to determine if the maximum number of alerts have been completed for the month. If this is true, then subroutine LVUPDT is called. The purpose of this subroutine will be discussed later. Upon return from LVUPDT the crew is placed in the appropriate inactive file, however, the transaction continues to flow through the EVENT

subroutine to trigger a search for the next day's alert crew.

If the maximum number of alerts has not been reached, the crew is checked for leave requests scheduled to start within the normal crew rest period. If a leave request is found, the crew schedule is updated and the crew is placed on the calendar to return on leave completion. This leave check immediately after an alert precludes the LEAVE subroutine from checking the event calendar for crews in crew rest status with concurrent leave requests. Again, the transaction also continues through the EVENT subroutine to trigger a search for the next day's alert crew.

If no leave is requested within two days of the alert, the crew is placed on the event calendar to return following crew rest. A 1.5-day delay is used to simulate the day of return from alert and the day of crew rest. This delay insures that a crew will not be selected for a "back-to-back" alert.

Finally, the crew transaction is used to trigger the search for the next day's alert crew for each LCC. A computed GO TO statement is used to call the appropriate next event subroutine. This call is based on the location of the completed alert.

Subroutine NXTALT

The NXTALT subroutine is used to select the next alert crew for non-SCP launch control centers (LCC). Crews are

A-21

selected by searching the EA queue (FILE 16) based on the priority ranking in the queue and the specified alert percentages. Following their selection, the crew is sent to the alert (queue) specified by IFILE.

Before attempting a crew search, the EA queue is checked to determine if any crews are available for alert duty. The variable NEXT is set equal to the pointer which identifies the first crew in the EA queue. If NEXT equals zero the queue is empty and a warning message is provided. In addition to the warning, a dummy transaction is placed on the event calendar to insure subsequent attempts to find an alert crew for that LCC.

If at least one crew is available for alert, the search begins by drawing a random number (RN) between zero and one. If RN is less than or equal to one of the specified percentages the search will proceed for the crew matching the required attribute(s). The first crew found to match these requirements will be removed from the EA queue and sent to the alert queue specified by IFILE. If no crew matching the attribute(s) is found, another search will be initiated to select the first line crew available. Finally, if this search fails, the first crew of any type will be selected. In all these searches, the first crew is determined by the priority ranking specified for the EA queue.

The searches are implemented by a repeat until type structure. Specifically, as each crew in the EA queue is checked the variable NEXT is set equal to the pointer of the

next crew in the file. If the correct crew is found then
the search will stop, otherwise the search will continue
until NEXT is equal to zero indicating all crews in the file
have been checked.

The purpose of the alert selection percentages are to
help spread the DOV/DOT and flight commander alerts through-
out the month and to control the number of integral alerts.
The goal of distributing these alerts throughout the month
is also assisted by searching for a line crew when the ini-
tial search fails.

Subroutine NXTSCP

The NXTSCP subroutine provides the same function as the
NXTALT subroutine but for SCP only alerts. The explanation
of NXTALT applies to NXTSCP with the following exceptions.
First, any crew selected must be SCP qualified (attribute 3
= 1). Second, if the first search is unsuccessful, the next
search will choose the first available SCP crew of any type.
Finally, although other crews may be available, a warning
message will be provided when an SCP qualified crew is not
available for an SCP alert.

Subroutine LEAVE

The LEAVE subroutine is used to remove crews from the EA
queue for their requested periods of leave. The subroutine
also updates the crew schedule with the leave code, 77, for

A-23

each day of leave. Crews removed from the EA queue are placed on the event calendar for the duration of their leave and return to the EA queue at leave completion.

The subroutine begins by scheduling the next leave check to occur one day later. Since the first leave check occurs at .8 days, the remaining checks are made at day 1.8, 2.8, etc. Using this method, leave requests are reviewed prior to determining the next day's alert crews.

The same basic search structure is used as in the next alert subroutines. The first crew's leave requests are copied into the LVBEG and LVEND arrays. These values are then checked to determine if leave is scheduled to start within the next two days. If leave is to start within this time, the crew schedule is updated and the duration of leave is determined for event calendar timing. The crew is then removed from the EA queue and placed on the event calendar with a return scheduled when the leave is completed. To preserve the pointer to the next crew in the EA queue, the variable MOVE is used to temporarily hold the next crew's position prior to removing the leave crew. When the transfer to the event calendar is complete, NEXT is set equal to MOVE and the search continues until all crews in the EA queue have been checked.

Subroutine LVUPDT

Since the LEAVE subroutine only searches the EA queue, only active crews are checked for leave requirements. The

LVUPDT subroutine is used to check crews who have reached

the maximum number of alerts and are about to be placed in

an inactive status. Because these crews have completed

their alert requirements for the month, only the crew sched-

ule must be updated.

The LVBEG and LVEND arrays are again used to hold leave

information. A simple loop is then used to check if any

leave is scheduled after the current day. If leave is

found, then the crew schedule is updated to reflect this

leave.

Subroutine OTPUT

The calling of the OTPUT subroutine by the SLAM proces-

sor signals the completion of the crew alert scheduling

simulation. At this point leave and alert requirements for

the month are complete and reflected in the crew schedule,

ISCHED. The OTPUT subroutine now calls the subroutines to

schedule all training requirements and gather statistics on

the monthly schedule.

Once all training has been scheduled and statistics com-

puted, OTPUT will print the crew scheduling array. The

codes used in the scheduling array are listed in Figure A-4.

To accommodate the different month lengths, a variable for-

mat structure is used. This variable formatting is accom-

plished using a character array (FORMAT) to store the dif-

ferent possibilities and the character variable FMT1.

Next, the statistical data for the month is printed.

```
┌─────────────────────────────────────┐
│           SCHEDULE CODES             │
│                                      │
│                                      │
│      Activity            Code        │
│                                      │
│                                      │
│      Alerts              1-15        │
│                                      │
│      Return Day          99          │
│                                      │
│      Leave               77          │
│                                      │
│      MPT Session         66          │
│                                      │
│      Classroom                       │
│      Training            XX          │
│                                      │
│                                      │
│      XX -- User Supplied Code        │
│                                      │
└─────────────────────────────────────┘
```

Figure A-4.   Crew Schedule Codes

First, information on the number of alerts, integral alerts,
   free days, and actual leave days is printed for each crew.
Cumulative mean and standard deviation statistics are then
printed for line crews covering the number of alerts, inte-
gral alerts, and free days.

Subroutine TRNING

   Subroutine TRNING assigns crews to classroom training
requirements. Up to three types of classroom training can
be accommodated. In addition, each type training can be of-
fered on a maximum of 15 days and the maximum (up to 25
crews) and minimum (1 crew) class sizes can be different for

each type training. Finally, a warning is printed for any crew not scheduled for a particular type of training and a recap of the actual class days and class sizes is provided.

Three subroutine peculiar arrays are needed in TRNING. Array ICOUNT stores the number of crews scheduled for each possible training day. Logical array NOSCHD keeps track of whether a crew has been scheduled for training or not. In NOSCHED .TRUE. indicates the crew has not been scheduled for training. Finally, array ICREW holds the crew numbers of crews selected for training on the first pass on a given training day.

The subroutine assigns crews using a two pass system. On the first pass up to 80 percent of the maximum class size is scheduled. If it was not possible to schedule 80 percent of the maximum class size, but the number of crews available was equal to at least the minimum class size, then these crews are scheduled. If neither of these conditions are satisfied, then no crews are scheduled for that day and the subroutine moves to the next training day. This process continues until all training days are considered.

The actual scheduling of crews for classroom training is accomplished by updating the crew scheduling array with the appropriate training code (MRKDAY). In addition, .TRUE. entries are changed to .FALSE. for all scheduled crews in array NOSCHD.

Next the second pass for assigning crews to classroom training is accomplished. Training days previously sched-

uled are now filled up to the maximum allowed class size. As crews are scheduled, the crew scheduling array, ISCHED, and the NOSCHD array are updated. Using the data contained in NOSCHD, any crew not scheduled for training is reported to the user.

Finally, a recap of the type classroom training scheduled is provided. This recap includes the type training (identified by the MRKDAY code), training date, and number crews assigned on that day. Additional calls are then made to this subroutine for each type classroom training requested.

Subroutine MPTTRN

The MPTTRN subroutine is used to schedule either one or two MPT sessions per crew. When more than one MPT session is requested, a minimum time between training sessions is specified to preclude back to back MPT training. All crews are considered for the MPT sessions and crews not assigned are reported to the user.

The arrays MPTSCH(i,j) and NOASGN are peculiar to this subroutine. MPTSCH(i,j) is used to store the MPT session date for each crew, i, and each MPT session number, j (1 or 2). The MPT session date is important when scheduling more than one training session to determine if the minimum time between MPTs (MTBMPT) requirement is satisfied. The array NOASGN simply keeps track of the number of MPT sessions assigned on each day. This number is then compared to the

number of sessions available in MPTDAY to prevent over
scheduling.

The subroutine considers each crew by searching the
entire month of available MPT sessions before moving to the
next crew. If an available period is found the crew is
scheduled by updating the crew schedule and the MPTSCH ar-
ray. In addition, if more than one MPT session was speci-
fied, only MPT sessions meeting the MTBMPT criteria will be
scheduled for the second session. All crews are checked for
first session availability before moving to the second ses-
sion request.

Any crews not scheduled for MPT training are reported to
the user. Crews are reported by both crew number and which
MPT session could not be scheduled. The conclusion of the
MPTTRN subroutine signals the completion of all crew sched-
uling activities.

Subroutine STATS

The STATS subroutine is used to collect individual crew
and cumulative line crew data from the monthly schedule.
The information is taken from the crew scheduling array,
ISCHED, which now contains all monthly requirements.

First the subroutine loops through the schedule collect-
ing information on each crew. This information includes the
number of alerts, number of integral alerts, and number of
free days which includes the number of actual leave days.

Next, a second looping is used to accumulate data on

line crews. The line crews are separated from the other
crews by the values stored in the INTFLT array. Following
this initial data collection, the mean and standard devia-
tion for each measure are computed.

PROGRAM LISTING: SLMWRT

```
      PROGRAM SLMWRT
C
C     **********************************************
C     *   THIS PROGRAM PROVIDES A USER FRIEND-  *
C     *   LY METHOD OF CREATING THE DATA FILE   *
C     *   REQUIRED BY THE SLAM CREW SCHEDULING  *
C     *   PROGRAM.  BY RESPONDING TO PROGRAM    *
C     *   PROMPTS, THE USER SPECIFIES INFORMA-  *
C     *   TION FOR THE SLAM CONTROL CARDS,      *
C     *   INDIVIDUAL CREW ATTRIBUTES AND        *
C     *   TRAINING REQUIREMENTS.  THIS DATA IS  *
C     *   THEN WRITTEN TO THE FILE 'SLAMIN'     *
C     *   FOR USE WITH THE SLAM CREW SCHEDULING *
C     *   PROGRAM.  FOR MORE DETAILED INFORMA-  *
C     *   TION, PLEASE REFER TO THE USER GUIDE. *
C     **********************************************
C
      CHARACTER QUEUE(15)*4,PRIO1*6,PRIO2*6,DATE*8
      DIMENSION STRTLV(3),ENDLV(3),ITRNDY(3,15),MPTDAY(31),ITYPE(3),MRKD
     1AY(3),MAXCLS(3),MINCLS(3)
C
C     INITIALIZE ARRAY 'QUEUE' WITH NAMES
C     REPRESENTING EACH LAUNCH CONTROL CENTER
C
      DATA QUEUE/'ALFA','BRAV','CHAR','DELT','ECHO','FOXT','GOLF','HOTE'
     1,'INDI','JULI','KILO','LIMA','MIKE','NOVE','OSCA'/
C
C     PROMPT USER FOR RUN DATE
C
      PRINT *,' ENTER DATE (EG,10/12/83)'
      PRINT *
      READ 50,DATE
50    FORMAT(A8)
C
C     PROMPT USER FOR PRIMARY AND TIE-BREAKER
C     PRIORITIES FOR THE CREW FILE
C
      PRINT *,' ENTER PRIORITY FOR CREW FILE (EG, LVF(5))'
      PRINT *
      READ 100,PRIO1
      PRINT *,' ENTER TIE-BREAKER PRIORITY'
      PRINT *
      READ 100,PRIO2
100   FORMAT(A6)
C
C     PROMPT USER FOR NUMBER OF DAYS IN THE MONTH
C
```

```
          PRINT X,' ENTER NUMBER OF DAYS IN MONTH'
          PRINT X
          READ X,NODAYS
C
C         PREPARE FILE 'SLAMIN' FOR THE NEW DATA
C
          OPEN(10,FILE='SLAMIN')
          REWIND 10
C
C         WRITE SLAM CONTROL CARDS TO FILE 'SLAMIN'
C         USING INFORMATION PROVIDED FROM ABOVE PROMPTS
C
          WRITE(10,X)'GEN,RNUSS,CREW SCHEDULE,',DATE,',1;'
          WRITE(10,X)'LIMITS,20,12,100;'
          WRITE(10,X)'PRIORITY/16,',PRIO1,'/NCLNR,',PRIO2,';'
          WRITE(10,X)'NETWORK;'
C
C         CONSTRUCT SLAMIN NETWORK OF 15 QUEUE NODES
C         REPRESENTING THE 15 LAUNCH CONTROL CENTERS.
C         LABELS FOR EACH QUEUE NODE PROVIDED BY
C         CHARACTER ARRAY 'QUEUE'. FOR MORE INFOR-
C         MATION, SEE USER GUIDE AND SLAM NETWORK DIAGRAM
C
          DO 1 I=1,15
             WRITE(10,X)QUEUE(I),'  QUEUE(',I,');'
             WRITE(10,X)'        ACTIVITY(1)/',I,',',1.0;'
             WRITE(10,X)'        EVENT,',I,';'
             WRITE(10,X)'        TERMINATE;'
1         CONTINUE
          WRITE(10,X)'        ENDNETWORK;'
          WRITE(10,X)'INIT,0.0,',REAL(NODAYS),';'
          WRITE(10,X)'FIN;'
C
C         SLAM NETWORK AND ASSOCIATED CONTROL CARDS
C         COMPLETE.  INITIALIZE CREW ATTRIBUTES FOR
C         NUMBER OF ALERTS AND SPARE ATTRIBUTE TO 0
C
          ALERTS=0.0
          SPARE=0.0
C
C         PROMPT USER FOR TOTAL NUMBER OF CREWS AND
C         VERIFY NUMBER OF CREWS 100 OR LESS, IF NOT,
C         PRINT WARNING AND REQUEST NEW INPUT
C
11        PRINT X,' ENTER NUMBER OF CREWS'
          PRINT X
          READ X,NOCRWS
          IF (NOCRWS.GT.100) THEN
             PRINT X,' WARNING: NUMBER OF CREWS MUST BE 100 OR LESS'
             GO TO 11
          ENDIF
          WRITE(10,X)NOCRWS
C
C         INFORM USER THAT FIRST FIFTEEN ENTRIES
```

```
C       REPRESENT THE ALERTS FOR DAY 1
C
        PRINT X,' FIRST FIFTEEN ENTRIES SENT ON ALERT FOR DAY 1'
C
C       BEGIN LOOP TO GATHER ATTRIBUTES FOR EACH CREW.
C       REINITIALIZE LEAVE DATES TO 0 BEFORE NEXT
C       CREW ENTRY
C
        DO 5 I=1,NOCRWS
           DO 10 J=1,3
              STRTLV(J)=0.0
              ENDLV(J)=0.0
10         CONTINUE
           PRINT X,' ENTRY NUMBER ',I
           PRINT X
C
C       PROMPT USER FOR IDENTIFYING DATA ON EACH CREW.
C       CHECK FOR IMPROPER DATA, PRINT WARNING(S)
C       AND REQUEST NEW INPUT
C
           PRINT X,' ENTER CREW NUMBER, FLIGHT, SCP QUAL, AND TYPE'
           PRINT X
           READ X,CRWNUM,FLTDES,SCPDES,CRWTYP
22         IF (CRWNUM.LE.0.0.OR.CRWNUM.GT.100.)THEN
              PRINT X,' WARNING: CREW NUMBER OUT OF RANGE'
              PRINT X,' ENTER CREW NUMBER (1-100)'
              PRINT X
              READ X,CRWNUM
              GO TO 22
           ENDIF
33         IF (FLTDES.LE.0.0.OR.FLTDES.GT.17.) THEN
              PRINT X,' WARNING: FLIGHT DESIGNATOR OUT FRNGE'
              PRINT X,' ENTER FLIGHT DESIGNATOR (1-17)'
              PRINT X
              READ X,FLTDES
              GO TO 33
           ENDIF
44         IF (SCPDES.LT.0.0.OR.SCPDES.GT.1.) THEN
              PRINT X,' WARNING: SCP QUALIFICATION OUT OF RANGE'
              PRINT X,' ENTER SCP QUALIFICATION (0 OR 1)'
              PRINT X
              READ X,SCPDES
              GO TO 44
           ENDIF
55         IF (CRWTYP.LT.0.0.OR.CRWTYP.GT.2.) THEN
              PRINT X,' WARNING: CREW TYPE OUT OF RANGE'
              PRINT X,' ENTER CREW TYPE (0,1,2)'
              READ X,CRWTYP
              GO TO 55
           ENDIF
C
C       PROMPT USER FOR LEAVE/NONAVAILABILITY DATES
C       CHECK LEAVE DATES FOR INCONSISTENT LEAVE
C       BEGIN AND END DATES
```

```
C
      PRINT X,' HOW MANY LEAVE PERIODS REQUESTED (0-3)'
         PRINT X
         READ X,LVREQ
         DO 15 J=1,LVREQ
66          PRINT X,' ENTER START AND END LEAVE DATES, PERIOD ',J
            PRINT X
         READ X,STRTLV(J),ENDLV(J)
      IF (ENDLV(J).LT.STRTLV(J)) THEN
         PRINT X,' WARNING: END LEAVE LESS THAN START LEAVE DATE'
         GO TO 66
      ENDIF
15    CONTINUE
C
C     PROMPT USER FOR ACTUAL NUMBER OF LEAVE DAYS
C     AS OPPOSED TO OTHER TYPES OF NONAVAILABILITY
C
      IF (LVREQ.NE.0) THEN
         PRINT X,'ENTER NUMBER OF ACTUAL LEAVE DAYS FOR CREW ',CRWNUM
         PRINT X
         READ X,LVDAYS
      ELSE
         LVDAYS=0
      ENDIF
C
C     WRITE INDIVIDUAL CREW DATA TO INPUT FILE 'SLAMIN'
C
      WRITE(10,X)CRWNUM,FLTDES,SCPDES,CRWTYP,ALERTS,SPARE,STRTLV(1),E
     1NDLV(1),STRTLV(2),ENDLV(2),STRTLV(3),ENDLV(3)
      WRITE(10,X)LVDAYS
5     CONTINUE
C
C     BEGIN TRAINING DATA COLLECTION
C     PROMPT USER FOR NUMBER OF TYPES OF CLASSROOM
C     TRAINING TO BE SCHEDULED
C
      PRINT X
      PRINT X,' ENTER THE NUMBER OF CLASSROOM-TYPE TRAINING (1,2,3)'
      PRINT X
      READ X,NUMTRN
      WRITE(10,X)NUMTRN
      DO 20 I=1,NUMTRN
         ITYPE(I)=I
C
C     PROMPT USER FOR NUMERIC CODE FOR EACH
C     TRAINING TYPE
C
      PRINT X,' ENTER TRAINING CODE FOR TYPE ',ITYPE(I)
      PRINT X
      READ X,MRKDAY(I)
C
C     PROMPT USER FOR MAX AND MIN CLASS SIZES
C     CHECK INPUT FOR CONSISTENCY
C
```

```
77         PRINT X,' ENTER MAXIMUM AND MINIMUM CLASS SIZE'
           PRINT X
           READ X,MAXCLS(I),MINCLS(I)
        IF (MINCLS(I).GT.MAXCLS(I)) THEN
           PRINT X,' WARNING: MINIMUM CLASS LARGER THEN MAXIMUM CLASS'
           GO TO 77
        ENDIF '
C
C       PROMPT USER FOR POSSIBLE DATES TO SCHEDULE
C       CLASS FOR EACH TYPE TRAINING.  FIFTEEN
C       POSSIBILITIES ARE ALLOWED
C
        DO 25 J=1,15
           PRINT X,' ENTER TRAINING DAY FOR TYPE ',ITYPE(I)
           PRINT X
           READ X,ITRNDY(I,J)
           WRITE(10,X)ITRNDY(I,J)
25      CONTINUE
        WRITE(10,X)MRKDAY(I)
        WRITE(10,X)MAXCLS(I),MINCLS(I)
20      CONTINUE
C
C       PROMPT USER FOR THE NUMBER OF MPT PERIODS
C       TO BE SCHEDULED FOR EACH CREW.  IF MORE
C       THAN 1 PERIOD REQUESTED, PROMPT USER FOR
C       MINIMUM TIME ALLOWED BETWEEN MPT PERIODS
C
        PRINT X
        PRINT X,' ENTER NUMBER OF MPT PERIODS PER CREW (1 OR 2)'
        PRINT X
        READ X,NUMMPT
        WRITE(10,X)NUMMPT
        IF (NUMMPT.GT.1) THEN
           PRINT X,' ENTER MINIMUM TIME BEWTEEN MPT PERIODS'
           PRINT X
           READ X,MTBMPT
           WRITE(10,X)MTBMPT
        ENDIF
C
C       PROMPT USER FOR NUMBER OF MPT PERIODS AVAILABLE
C       ON EACH DAY OF THE MONTH
C
        PRINT X,' ENTER NUMBER OF MPT PERIDS PER DAY'
        PRINT X
        WRITE(10,X)NODAYS
        DO 30 I=1,NODAYS
           PRINT X,' DAY ',I
           PRINT X
           READ X,MPTDAY(I)
           WRITE(10,X)MPTDAY(I)
30      CONTINUE
C
C       PROMPT USER FOR MAXIMUM ALLOWABLE ALERTS
C       FOR EACH CREW TYPE
```

```
C
          PRINT *,' ENTER MAXIMUM NUMBER OF ALERTS FOR:'
          PRINT *,' LINE CREWS, FLT COMMANDERS, AND DOV/DOT'
          PRINT *
          READ *,MAXLIN,MAXFCC,MAXSHP
          WRITE(10,*)MAXLIN,MAXFCC,MAXSHP
C
C         PROMPT USER FOR ALERT ASSIGNMENT PERCENTAGES
C         (REFERENCE TO USER MANUAL REQUIRED TO
C         UNDERSTAND SIGNIFICANCE OF THESE SETTINGS)
C         CHECK PERCENTAGES FOR CONSISTENCY
C
          PRINT *,' ENTER PERCENTAGES FOR ALERT ASSIGNMENTS:'
88        PRINT *,' FOR NON-SCP ALERTS -- DOV/DOT, FLT CC, INTEGRAL'
          PRINT *
          READ *,SHPPCT,FCCPCT,FLTPCT
          IF (FLTPCT.LT.FCCPCT.OR.FCCPCT.LT.SHPPCT) THEN
             PRINT *,' WARNING:  INCONSISTENT DATA, REINPUT ALERT %'
             GO TO 88
          ENDIF
99        PRINT *,' FOR SCP ALERTS -- DOV/DOT, INTEGRAL'
          PRINT *
          READ *,SHPCTS,FTPCTS
          IF (FTPCTS.LT.SHPCTS) THEN
             PRINT *,' WARNING: INCONSISTENT DATA, REINPUT ALERT %'
             GO TO 99
          ENDIF
          WRITE(10,*)SHPPCT,FCCPCT,FLTPCT,SHPCTS,FTPCTS
C
C         PROMPT USER FOR DESIRED RANDOM NUMBER STREAMS
C
          PRINT *,'ENTER THE 2 RANDOM NUMBER STREAMS (1-10)'
          PRINT *
          READ *,ISTRM1,ISTRM2
          WRITE(10,*)ISTRM1,ISTRM2
C
C         CLOSE DATA FILE 'SLAMIN'.  CREW DATA ENTRY COMPLETE
C
          CLOSE(10)
          END
```

LIST OF VARIABLES: SLMWRT

ARRAYS:

ENDLV(3) - crew requested end leave dates (attributes 8, 10, 12); real

ITRNDY(3,15) - available training days for each type of classroom
training; integer

A-36

ITYPE(3) - identifies type of classroom training; integer

MAXCLS(3) - maximum class size for each type classroom training; integer

MINCLS(3) - minimum class size for each type classroom training; integer

MPTDAY(31) - number of available MPT sessions per day; integer

MRKDAY(3) - numeric code of each type classroom training; integer

QUEUE(15) - labels for SLAM network queues; character

STRTLV(3) - crew requested end leave dates (attributes 7, 9, 11); real

VARIABLES:

ALERTS - represents attribute 5, number of alerts, for individual crew information; real

CRWNUM - represents attribute 1, crew number, for individual crew information; real

CRWTYP - represents attribute 4, crew type, for individual crew information; real

DATE - date of run on SLAM control card; character

FCCPCT - percentage of flight commanders sent to non-SCP alerts; real

FLTDES - represents attribute 2, flight designator, for individual crew information; real

FLTPCT - percentage of crews sent to their assigned flight for non-SCP alerts; real

FTPCTS - percentage of crews sent to their assigned flight for SCP alerts; real

ISTRM1 - random number stream used for non-SCP alerts; integer

ISTRM2 - random number stream used for SCP alerts; integer

LVDAYS - number of actual leave days for each crew; integer

LVREQ - number of leave periods requested by crew; integer

MAXFCC - maximum number of alerts for flight commander; integer

MAXLIN - maximum number of alerts for line crews; integer

MAXSHP - maximum number of alerts for DOV/DOT crews; integer

MTBMPT - minimum time between MPT training; integer

NOCRWS - total number of crews; integer

NODAYS - number of days in month; integer

NUMMPT - number of MPT sessions requested per crew; integer

NUMTRN - number of classroom type training; integer

PRIO1 - primary priority for EA queue; character

PRIO2 - tie-breaker priority for EA queue; character

SCPDES - represents attribute 3, scp qualification, for individual crew
information; real

SHPCTS - percentage of DOV/DOT crews sent to SCP alerts; real

SHPPCT - percentage of DOV/DOT crews sent to non-SCP alerts; real

SPARE - represents spare attribute 6 for individual crew information;
real

PROGRAM LISTING: SLAM CONTROL CARDS


```
GEN,RNUSS,CREW SCHEDULE,01/27/84,1;
LIMITS,20,12,100;
PRIORITY/16,LVF(5),/NCLNR,LVF(4);
NETWORK;
ALFA  QUEUE(1);
      ACTIVITY(1)/1,1.0;
      EVENT,1;
      TERMINATE;
BRAV  QUEUE(2);
      ACTIVITY(1)/2,1.0;
      EVENT,2;
      TERMINATE;
CHAR  QUEUE(3);
      ACTIVITY(1)/3,1.0;
      EVENT,3;
      TERMINATE;
DELT  QUEUE(4);
      ACTIVITY(1)/4,1.0;
      EVENT,4;
      TERMINATE;
ECHO  QUEUE(5);
      ACTIVITY(1)/5,1.0;
      EVENT,5;
      TERMINATE;
```

```
FOXT  QUEUE(6);
      ACTIVITY(1)/6,1.0;
      EVENT,6;
      TERMINATE;
GOLF  QUEUE(7);
      ACTIVITY(1)/7,1.0;
      EVENT,7;
      TERMINATE;
HOTE  QUEUE(8);
      ACTIVITY(1)/8,1.0;
      EVENT,8;
      TERMINATE;
INDI  QUEUE(9);
      ACTIVITY(1)/9,1.0;
      EVENT,9;
      TERMINATE;
JULI  QUEUE(10);
      ACTIVITY(1)/10,1.0;
      EVENT,10;
      TERMINATE;
KILO  QUEUE(11);
      ACTIVITY(1)/11,1.0;
      EVENT,11;
      TERMINATE;
LIMA  QUEUE(12);
      ACTIVITY(1)/12,1.0;
      EVENT,12;
      TERMINATE;
MIKE  QUEUE(13);
      ACTIVITY(1)/13,1.0;
      EVENT,13;
      TERMINATE;
NOVE  QUEUE(14);
      ACTIVITY(1)/14,1.0;
      EVENT,14;
      TERMINATE;
OSCA  QUEUE(15);
      ACTIVITY(1)/15,1.0;
      EVENT,15;
      TERMINATE;
      ENDNETWORK;
INIT,0.0,30.;
FIN;
```

```
          PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7,TAPE15)
C
C     *****************************************
C     *   PROGRAM 'MAIN' IS USED TO DIMENSION  *
C     *   SLAM STORAGE ARRAYS, SPECIFY INPUT    *
C     *   AND OUTPUT PARAMETERS AND CALL THE    *
C     *   SLAM SUBROUTINE WHICH SUPPLIES EXEC-  *
C     *   UTIVE CONTROL FOR THE SIMULATION      *
C     *****************************************
C
          DIMENSION NSET(5000)
          COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
         1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
          COMMON QSET(5000)
          EQUIVALENCE(NSET(1),QSET(1))
          NNSET=5000
          NCRDR=5
          NPRNT=6
          NTAPE=7
          CALL SLAM
          STOP
          END
C
C     *********************************************
C     *   SUBROUTINE 'INTLC' IS USED TO INI-       *
C     *   TIALIZE PROGRAM VARIABLES AND READ        *
C     *   CREW AND TRAINING INFORMATION FROM        *
C     *   THE DATA FILE 'SLAMIN'                     *
C     *********************************************
C
          SUBROUTINE INTLC
          COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
         1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
          COMMON/UCOM1/IDAY,NXTDAY,IFILE,ISCHED(100,32),DUMMY(14),LVBEG(3),L
         1VEND(3),INTFLT(100),NOCRWS,NODAYS,NUMTRN,MAXLIN,MAXFCC,MAXSHP
          COMMON/UCOM2/ITRNDY(3,15),ITYPE(3),MRKDA(),MPTDAY(31),NUMMPT,MTB
         1MPT,N,MAXCLS(3),MINCLS(3)
          COMMON/UCOM3/NOALTS(100),FREDAY(100),INTALT(100),LVDAYS(100)
          COMMON/UCOM5/SHPPCT,FCCPCT,FLTPCT,SHPCTS,FTPCTS,ISTRM1,ISTRM2
          DATA ITRNDY/45*0/
          DATA DUMMY/14*0.0/
          DATA LVBEG/3*0/
          DATA LVEND/3*0/
          DATA MPTDAY/31*6/
          DATA INTFLT/100*0/
          DO 1 I=1,100
             DO 2 J=1,32
             ISCHED(I,J)=0
2            CONTINUE
1         CONTINUE
          IFILE=0
```

```
      IDAY=0
      NXTDAY=0
      READ(NCRDR,X)NOCRWS
C
C     CREW ATTRIBUTES AND LEAVE DAYS ARE READ IN
C     FOR EACH CREW.  FROM THIS DATA THE ARRAY
C     'INTFLT' IS CREATED AND CONTAINS THE
C     ASSIGNED FLIGHT FOR EACH CREW WITH FLIGHT
C     COMMANDERS IDENTIFIED BY A NEGATIVE NUMBER.
C     THIS ARRAY WILL BE USED TO EASE DATA COLLECTION
C     IN SUBROUTINE 'STATS'
C
C     NOTE THAT THE FIRST 15 ENTRIES ARE SENT TO
C     THE ALERT QUEUES TO SIMULATE THE FIRST DAY
C     OF THE ALERT SCHEDULE.  FOLLOWING THESE
C     FIRST 15 ENTRIES, THE REMAINDER OF THE CREWS
C     ARE SENT TO THE EA QUEUE (FILE 16)
C     OR IF LEAVE IS SCHEDULED TO BEGIN ON THE
C     FIRST DAY, PLACED ON THE EVENT CALENDAR
C     TO RETURN AT COMPLETION OF THE LEAVE
C
      DO 5 I=1,15
         READ(NCRDR,X)(ATRIB(J),J=1,12)
         READ(NCRDR,X)LVDAYS(NINT(ATRIB(1)))
         IF (NINT(ATRIB(4)).EQ.1) THEN
            INTFLT(NINT(ATRIB(1)))=-(NINT(ATRIB(2)))
         ELSE
            INTFLT(NINT(ATRIB(1)))=NINT(ATRIB(2))
         ENDIF
         CALL FILEM(I,ATRIB)
5     CONTINUE
      DO 10 I=16,NOCRWS
         READ(NCRDR,X)(ATRIB(J),J=1,12)
         READ(NCRDR,X)LVDAYS(NINT(ATRIB(1)))
         IF (NINT(ATRIB(4)).EQ.1) THEN
            INTFLT(NINT(ATRIB(1)))=-(NINT(ATRIB(2)))
         ELSE
            INTFLT(NINT(ATRIB(1)))=NINT(ATRIB(2))
         ENDIF
         IF (NINT(ATRIB(7)).EQ.1) THEN
            DO 12 J=NINT(ATRIB(7)),NINT(ATRIB(8))
               ISCHED(NINT(ATRIB(1)),J)=77
12          CONTINUE
            DURLV=ATRIB(8)-ATRIB(7)+1.2
            CALL SCHDL(25,DURLV,ATRIB)
            GO TO 10
         ENDIF
         CALL FILEM(16,ATRIB)
1     CONTINUE
C
C     TRAINING INFORMATION FOR BOTH CLASSROOM
C     AND MPT PERIODS IS READ FROM THE INPUT FILE
C
      READ(NCRDR,X)NUMTRN
```

```
          DO 15 I=1,NUMTRN
             DO 20 J=1,15
             READ(NCRDR,X)ITRNDY(I,J)
20           CONTINUE
          READ(NCRDR,X)MRKDAY(I)
          READ(NCRDR,X)MAXCLS(I),MINCLS(I)
          ITYPE(I)=I
15        CONTINUE
          READ(NCRDR,X)NUMMPT
          IF (NUMMPT.GT.1) THEN
             READ(NCRDR,X)MTBMPT
          ELSE
             MTBMPT=0
          ENDIF
          READ(NCRDR,X)NODAYS
          DO 25 I=1,NODAYS
             READ(NCRDR,X)MPTDAY(I)
25        CONTINUE
C
C         READ MAXIMUM NUMBER OF ALERTS FOR EACH CREW
C         CATEGORY, INTERNAL ALERT PERCENTAGES AND
C         SLAM RANDOM NUMBER STREAMS TO BE USED
C
          READ(NCRDR,X)MAXLIN,MAXFCC,MAXSHP
          READ(NCRDR,X)SHPPCT,FCCPCT,FLTPCT,SHPCTS,FTPCTS
          READ(NCRDR,X)ISTRM1,ISTRM2
C
C         THIS CALL TO THE SLAM SUBROUTINE 'SCHDL'
C         IS USED TO PLACE THE TIME TO START THE FIRST
C         LEAVE SEARCH ON THE EVENT CALENDAR.
C         ARRAY 'DUMMY' IS USED SINCE THIS EVENT
C         DOES NOT CONCERN A SPECIFIC CREW
C
          CALL SCHDL(30,0.8,DUMMY)
          RETURN
          END
C
C         XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C         X   SUBROUTINE 'EVENT' IS USED TO DIRECT X
C         X   PROGRAM ACTIONS DURING THE ALERT     X
C         X   SCHEDULING PHASE OF THE SIMULATION.  X
C         X   TRANSACTIONS ARRIVING AT THIS SUB-   X
C         X   ROUTINE CARRY A CODE (IFN) WHICH     X
C         X   DETERMINES THEIR DISPOSITION         X
C         XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C
          SUBROUTINE EVENT(IFN)
          COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
         1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
          COMMON/ICOM1/IDAY,NXTDAY,IFILE,ISCHED(100,32),DUMMY(14),LVBEG(3),L
         1VEND(3),INTFLT(100),NOCRWS,NODAYS,NUMTRN,MAXLIN,MAXFCC,MAXSHP
          COMMON/UCOM5/SHPPCT,FCCPCT,FLTPCT,SHPCTS,FTPCTS,ISTRM1,ISTRM2
          IFILE=IFN
          IDAY=NINT(TNOW)
```

```fortran
          NXTDAY=IDAY+1
C
C      IFN OF 25 INDICATES A CREW IS RETURNING
C      FROM CREW REST OR LEAVE.  CREW IS NOW SENT
C      TO THE EA QUEUE
C
       IF (IFN.EQ.25) THEN
          CALL FILEM(16,ATRIB)
          RETURN
C
C      IFN OF 30 INDICATES TIME TO SEARCH CREW FOR
C      LEAVE REQUESTS
C
       ELSE IF (IFN.EQ.30) THEN
          CALL LEAVE
          RETURN
       ENDIF
C
C      ATRIB(1) SET TO ZERO INDICATES DUMMY CREW
C      THE DUMMY CREW TRANSACTION IS USED WHEN NO
C      CREWS WERE AVAILABLE THE PREVIOUS DAY FOR
C      THAT LCC.  THIS TRANSACTION IS THEN SENT TO
C      STATEMENT 50 FOR TRANSFER TO THE CORRECT
C      NEXT ALERT ROUTINE
C
       IF (NINT(ATRIB(1)).EQ.0) GO TO 50
C
C      UPDATE THE CREW SCHEDULE, ARRAY 'ISCHED',
C      WITH THE COMPLETED ALERT.  CODE 99 IS USED
C      TO INDICATE THE DAY OF RETURN FROM ALERT.
C      ALSO, THE NUMBER OF ALERTS (ATRIB(5)) IS
C      INCREASED BY 1
C
       ISCHED(NINT(ATRIB(1)),IDAY)=IFILE
       ISCHED(NINT(ATRIB(1)),(IDAY+1))=99
       ATRIB(5)=ATRIB(5)+1.0
C
C      CHECK CREW COMPLETING ALERT FOR MAXIMUM NUMBER
C      OF ALERTS.  IF MAXIMUM REACHED CHECK CREW FOR
C      LEAVE REQUESTS FOR THE REMAINDER OF THE MONTH
C      AND FILE CREW IN APPLICABLE MAX ALERT FILE
C
       IF (NINT(ATRIB(4)).EQ.0 .AND. NINT(ATRIB(5)).GE.MAXLIN) THEN
          CALL LVUPDT
          CALL FILEM(17,ATRIB)
       ELSE IF (NINT(ATRIB(4)).EQ.1 .AND. NINT(ATRIB(5)).GE.MAXFCC) THEN
          CALL LVUPDT
          CALL FILEM(19,ATRIB)
       ELSE IF (NINT(ATRIB(4)).EQ.2 .AND. NINT(ATRIB(5)).GE.MAXSHP) THEN
          CALL LVUPDT
          CALL FILEM(18,ATRIB)
C
C      CHECK OTHER RETURNING CREWS FOR LEAVE IMMEDIATELY
C      FOLLOWING RETURN FROM ALERT.  IF FOUND, UPDATE
```

```
C       CREW SCHEDULE AND PLACE ON EVENT CALENDAR FOR
C       LEAVE RETURN
C
        ELSE IF (NINT(ATRIB(7)).EQ.(NXTDAY+1)) THEN
           DO 75 J=NINT(ATRIB(7)),NINT(ATRIB(8))
              ISCHED(NINT(ATRIB(1)),J)=77
75         CONTINUE
           DURLV=ATRIB(8)-ATRIB(7)+2.2
           CALL SCHDL(25,DURLV,ATRIB)
        ELSE IF (NINT(ATRIB(9)).EQ.(NXTDAY+1)) THEN
           DO 76 J=NINT(ATRIB(9)),NINT(ATRIB(10))
              ISCHED(NINT(ATRIB(1)),J)=76
76         CONTINUE
           DURLV=ATRIB(10)-ATRIB(9)+2.2
           CALL SCHDL(25,DURLV,ATRIB)
        ELSE IF (NINT(ATRIB(11)).EQ.NTDY+1) THEN
           DO 77 J=NINT(ATRIB(11)),NINT(ATRIB(12))
              ISCHED(NINT(ATRIB(1)),J)=76
77         CONTINUE
           DURLV=ATRIB(12)-ATRIB(11)+2.2
           CALL SCHDL(25,DURLV,ATRIB)
C
C       FOR ALL OTHER CREWS COMPLETING ALERT, PLACE
C       CREW ON EVENT CALENDAR TO RETURN TO THE
C       EA QUEUE FOLLOWING CREW REST PERIOD
C
        ELSE
           CALL SCHDL(25,1.5,ATRIB)
        ENDIF
C
C       BASED ON LOCATION OF JUST COMPLETED ALERT,
C       CALL NEXT ALERT SUBROUTINE TO SCHEDULE THE
C       NEXT DAY ALERT CREW
C
50      GO TO(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15),IFN
1       CALL NXTSCP
        RETURN
2       CALL NXTALT
        RETURN
3       CALL NXTALT
        RETURN
4       CALL NXTALT
        RETURN
5       CALL NXTALT
        RETURN
6       CALL NXTSCP
        RETURN
7       CALL NXTALT
        RETURN
8       CALL NXTALT
        RETURN
9       CALL NXTALT
        RETURN
10      CALL NXTALT
```

```
         RETURN
11       CALL NXTSCP
         RETURN
12       CALL NXTALT
         RETURN
13       CALL NXTALT
         RETURN
14       CALL NXTALT
         RETURN
15       CALL NXTALT
         RETURN
         END
C
C        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C        X   SUBROUTINE 'NXTALT' IS USED TO       X
C        X   SCHEDULE THE ALERT CREWS FOR ALL     X
C        X   NON-SCP LAUNCH CONTROL CENTERS       X
C        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C
         SUBROUTINE NXTALT
         COMMON/SCOM1/ATRIB(100),D10),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
        1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
         COMMON/UCOM1/IDAY,NXTDAY,IFILE,ISCHED(100,32),DUMMY(14),LVBEG(3),L
        1VEND(3),INTFLT(100),NOCRWS,NODAYS,NUMTRN,MAXLIN,MAXFCC,MAXSHP
         COMMON/UCOM5/SHPPCT,FCCPCT,FLTPCT,SHPCTS,FTPCTS,ISTRM1,ISTRM2
C
C        THIS SETS THE VARIABLE 'NEXT' EQUAL TO THE
C        FIRST CREW IN THE EA QUEUE.  IF 'NEXT'
C        EQUALS 0, THEN THE CREW FILE IS EMPTY.  IN
C        THIS CASE A WARNING MESSAGE IS PRINTED FOR
C        THE DAY AND APPLICABLE LCC.  A DUMMY
C        TRANSACTION IS THEN USED TO INSURE A RETURN
C        TO THIS SUBROUTINE FOR THE FOLLOWING DAYS
C        ALERT SCHEDULING
C
         NEXT=MMFE(16)
         IF (NEXT.EQ.0) THEN
            WRITE(6,X)' CREW FILE EMPTY ON DAY ',NXTDAY,' FOR LCC ',IFILE
            CALL SCHDL(IFILE,1.0,DUMMY)
            RETURN
         ENDIF
C
C        THE EA QUEUE IS SEARCHED FOR A SPECIFIC CREW
C        BASED ON THE RANDOM NUMBER DRAWN AND THE
C        SPECIFIED PERCENTAGES
C
         RN=DRAND(ISTRM1)
10       CALL COPY(-NEXT,16,ATRIB)
         IF (RN.LE.SHPPCT) THEN
            IF (NINT(ATRIB(4)).EQ.2) THEN
               CALL RMOVE(-NEXT,16,ATRIB)
               CALL FILEM(IFILE,ATRIB)
               RETURN
            ENDIF
```

```
                  ELSE IF (RN.LE.FCCPCT) THEN
                      IF (NINT(ATRIB(4)).EQ.1 .AND. NINT(ATRIB(2)).EQ.IFILE) THEN
                          CALL RMOVE(-NEXT,16,ATRIB)
                          CALL FILEM(IFILE,ATRIB)
                          RETURN
                      ENDIF
                  ELSE IF (RN.LE.FLTPCT) THEN
                      IF (NINT(ATRIB(2)).EQ.IFILE) THEN
                          CALL RMOVE(-NEXT,16,ATRIB)
                          CALL FILEM(IFILE,ATRIB)
                          RETURN
                      ENDIF
                  ELSE
                      IF (NINT(ATRIB(4)).EQ.0) THEN
                          CALL RMOVE(-NEXT,16,ATRIB)
                          CALL FILEM(IFILE,ATRIB)
                          RETURN
                      ENDIF
                  ENDIF
                  NEXT=NSUCR(NEXT)
                  IF (NEXT.NE.0) GO TO 10
C
C        IF A CREW WITH THE REQUESTED ATTRIBUTES IS
C        NOT FOUND, THEN SEARCH FILE FOR LINE CREW
C
                  NEXT=MMFE(16)
20                CALL COPY(-NEXT,16,ATRIB)
                  IF (NINT(ATRIB(4)).EQ.0) THEN
                      CALL RMOVE(-NEXT,16,ATRIB)
                      CALL FILEM(IFILE,ATRIB)
                      RETURN
                  ENDIF
                  NEXT=NSUCR(NEXT)
                  IF (NEXT.NE.0) GO TO 20
C
C        IF LINE CREW NOT AVAILABLE, SELECT FIRST
C        CREW IN FILE
C
                  CALL RMOVE(-MMFE(16),16,ATRIB)
                  CALL FILEM(IFILE,ATRIB)
                  RETURN
                  END
C
C        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C        X    SUBROUTINE 'NXTSCP' IS USED TO       X
C        X    SCHEDULE THE ALERT CREWS FOR ALL     X
C        X    SCP LAUNCH CONTROL CENTERS           X
C        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C
                  SUBROUTINE NXTSCP
                  COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
                 1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
                  COMMON/UCOM1/IDAY,NXTDAY,IFILE,ISCHED(100,32),DUMMY(14),LVBEG(3),L
                 1VEND(3),INTFLT(100),NOCRWS,NODAYS,NUMTRN,MAXLIN,MAXFCC,MAXSHP
```

```
          COMMON/UCOM5/SHPPCT,FCCPCT,FLTPCT,SHPCTS,FTPCTS,ISTRM1,ISTRM2
C
C         THIS SETS THE VARIABLE 'NEXT' EQUAL TO THE
C         FIRST CREW IN THE EA QUEUE.  IF 'NEXT'
C         EQUALS 0, THEN THE CREW FILE IS EMPTY.  IN
C         THIS CASE A WARNING MESSAGE IS PRINTED FOR
C         THE DAY AND APPLICABLE SCP.  A DUMMY
C         TRANSACTION IS THEN USED TO INSURE A RETURN
C         TO THIS SUBROUTINE FOR THE FOLLOWING DAYS
C         ALERT SCHEDULING
C
          NEXT=MMFE(16)
          IF (NEXT.EQ.0) THEN
             WRITE(6,X)' CREW FILE EMPTY ON DAY ',NXTDAY
             CALL SCHDL(IFILE,1.0,DUMMY)
             RETURN
          ENDIF
C
C         THE EA QUEUE IS SEARCHED FOR A SPECIFIC SCP
C         CREW BASED ON THE RANDOM NUMBER DRAWN AND
C         THE SPECIFIED PERCENTAGES
C
          RN=DRAND(ISTRM2)
10        CALL COPY(-NEXT,16,ATRIB)
          IF (RN.LE.SHPCTS) THEN
             IF (NINT(ATRIB(4)).EQ.2 .AND. NINT(ATRIB(3)).EQ.1) THEN
                CALL RMOVE(-NEXT,16,ATRIB)
                CALL FILEM(IFILE,ATRIB)
                RETURN
             ENDIF
          ELSE IF (RN.LE.FTPCTS) THEN
             IF (NINT(ATRIB(2)).EQ.IFILE .AND. NINT(ATRIB(3)).EQ.1) THEN
                CALL RMOVE(-NEXT,16,ATRIB)
                CALL FILEM(IFILE,ATRIB)
                RETURN
             ENDIF
          ELSE
             IF (NINT(ATRIB(3)).EQ.1 .AND. NINT(ATRIB(4)).LT.2) THEN
                CALL RMOVE(-NEXT,16,ATRIB)
                CALL FILEM(IFILE,ATRIB)
                RETURN
             ENDIF
          ENDIF
          NEXT=NSUCR(NEXT)
          IF (NEXT.NE.0) GO TO 10
C
C         IF A CREW WITH THE REQUESTED ATTRIBUTES IS
C         NOT FOUND, THEN SEARCH FILE FOR ANY SCP CREW
C         IF NO SCP CREWS AVAILABLE PRINT WARNING MESSAGE
C         AND SCHEDULE DUMMY TRANSACTION
C
          NEXT=MMFE(16)
20        IF (NEXT.EQ.0) THEN
             WRITE(6,X)' NO SCP CREW FOR SCP ',IFILE,' ON DAY ',NXTDAY
```

```
              CALL SCHDL(IFILE,1.0,DUMMY)
              RETURN
          ENDIF
          CALL COPY(-NEXT,16,ATRIB)
          IF (NINT(ATRIB(3)).EQ.1) THEN
              CALL RMOVE(-NEXT,16,ATRIB)
              CALL FILEM(IFILE,ATRIB)
              RETURN
          ENDIF
          NEXT=NSUCR(NEXT)
          GO TO 20
          END
C
C     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C     X  SUBROUTINE 'LEAVE' CHECKS THE EA      X
C     X  QUEUE EACH DAY PRIOR TO SCHEDUL-      X
C     X  ING ALERTS FOR CREW LEAVE REQUESTS.   X
C     X  CREWS REQUESTING LEAVE WITHIN THE     X
C     X  NEXT TWO DAYS ARE REMOVED FROM THE    X
C     X  EA QUEUE (FILE 16) AND PLACED ON THE  X
C     X  EVENT CALENDAR FOR RETURN FOLLOWING   X
C     X  LEAVE COMPLETION                      X
C     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C
      SUBROUTINE LEAVE
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/UCOM1/IDAY,NXTDAY,IFILE,ISCHED(100,32),DUMMY(14),LVBEG(3),L
     1VEND(3),INTFLT(100),NOCRWS,NODAYS,NUMTRN,MAXLIN,MAXFCC,MAXSHP
C
C     SCHEDULE THE NEXT LEAVE SEARCH TO OCCUR THE
C     NEXT DAY.  BEGIN SEARCH OF THE EA QUEUE
C     AND COPY LEAVE START AND END DATES FOR EACH
C     CREW
C
      CALL SCHDL(30,1.0,DUMMY)
      NEXT=MMFE(16)
10    IF (NEXT.EQ.0) RETURN
      CALL COPY(-NEXT,16,ATRIB)
      LVBEG(1)=NINT(ATRIB(7))
      LVBEG(2)=NINT(ATRIB(9))
      LVBEG(3)=NINT(ATRIB(11))
      LVEND(1)=NINT(ATRIB(8))
      LVEND(2)=NINT(ATRIB(10))
      LVEND(3)=NINT(ATRIB(12))
C
C     IF LEAVE START WITHIN THE NEXT TWO DAYS FOUND
C     REMOVE CREW AND PLACE ON EVENT CALENDAR FOR
C     RETURN FOLLOWING LEAVE COMPLETION.  IN
C     ADDITION, UPDATE SCHEDULING ARRAY.  (NOTE:
C     VARIABLE 'MOVE' USED TO HOLD POINTER POSITION
C     WHEN CREW REMOVED FROM THE EA QUEUE)
C
      DO 20 K=1,3
```

```
          IF (LVBEG(K).EQ.NXTDAY) THEN
              DO 1 J=LVBEG(K),LVEND(K)
                  ISCHED(NINT(ATRIB(1)),J)=77
    1         CONTINUE
              DURLV=REAL(LVEND(K)-LVBEG(K))+1.2
              MOVE=NSUCR(NEXT)
              CALL RMOVE(-NEXT,16,ATRIB)
              CALL SCHDL(25,DURLV,ATRIB)
              NEXT=MOVE
              GO TO 10
          ENDIF
          IF (LVBEG(K).EQ.(NXTDAY+1)) THEN
              DO 5 J=LVBEG(K),LVEND(K)
                  ISCHED(NINT(ATRIB(1)),J)=77
    5         CONTINUE
              DURLV=REAL(LVEND(K)-LVBEG(K))+2.2
              MOVE=NSUCR(NEXT)
              CALL RMOVE(-NEXT,16,ATRIB)
              CALL SCHDL(25,DURLV,ATRIB)
              NEXT=MOVE
              GO TO 10
          ENDIF
   20     CONTINUE
          NEXT=NSUCR(NEXT)
          GO TO 10
          END
    C
    C     ************************************************
    C     *   SUBROUTINE 'LVUPDT' IS USED TO            *
    C     *   UPDATE THE CREW SCHEDULING ARRAY          *
    C     *   FOR THOSE CREWS WHO HAVE REACHED          *
    C     *   THEIR MAXIMUM ALERT LOAD BUT STILL        *
    C     *   HAVE LEAVE REQUESTS FOR THE REMAIN-       *
    C     *   DER OF THE MONTH.                         *
    C     ************************************************
    C
          SUBROUTINE LVUPDT
          COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
         1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
          COMMON/UCOM1/IDAY,NXTDAY,IFILE,ISCHED(100,32),DUMMY(14),LVBEG(3),L
         1VEND(3),INTFLT(100),NOCRWS,NODAYS,NUMTRN,MAXLIN,MAXFCC,MAXSHP
    C
    C     COPY LEAVE INFORMATION FROM CREW ATIBUTES AND
    C     UPDATE CREW SCHEDULING ARRAY IF REQUIRED
    C
          LVBEG(1)=NINT(ATRIB(7))
          LVBEG(2)=NINT(ATRIB(9))
          LVBEG(3)=NINT(ATRIB(11))
          LVEND(1)=NINT(ATRIB(8))
          LVEND(2)=NINT(ATRIB(10))
          LVEND(3)=NINT(ATRIB(12))
          DO 5 K=1,3
          IF (LVBEG(K).GE.IDAY) THEN
              DO 1 J=LVBEG(K),LVEND(K)
```

```fortran
                ISCHED(NINT(ATRIB(1)),J)=77
1         CONTINUE
        ENDIF
5       CONTINUE
        RETURN
        END
C
C       **********************************************
C       *  SUBROUTINE 'OTPUT' IS CALLED BY SLAM  *
C       *  WHEN THE ALERT SCHEDULING SIMULATION  *
C       *  IS COMPLETE.  THE SUBROUTINES FOR     *
C       *  ASSIGNING TRAINING DAYS AND STATIS-   *
C       *  TICS COLLECTION ARE THEN CALLED FROM  *
C       *  'OTPUT'.  FINALLY, TRAINING DATA,     *
C       *  THE CREW SCHEDULE, AND MONTHLY STA-   *
C       *  TISTICS ARE PRINTED                   *
C       **********************************************
C
        SUBROUTINE OTPUT
        COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
       1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
        COMMON/UCOM1/IDAY,NXTDAY,IFILE,ISCHED(100,32),DUMMY(14),LVBEG(3),L
       1VEND(3),INTFLT(100),NOCRWS,NODAYS,NUMTRN,MAXLIN,MAXFCC,MAXSHP
        COMMON/UCOM2/ITRNDY(3,15),ITYPE(3),MRKDAY(3),MPTDAY(31),NUMMPT,MTB
       1MPT,N,MAXCLS(3),MINCLS(3)
        COMMON/UCOM3/NOALTS(100),FREDAY(100),INTALT(100),LVDAYS(100)
        COMMON/UCOM4/ALTMN,ALTSTD,ALTIMN,ALTIST,FREMN,FRESTD
        CHARACTER FORMAT(4)*14,FMT1*14
C
C       CALL THE 'TRNING' SUBROUTINE FOR EACH TYPE
C       OF CLASSROOM TRAINING.  CALL THE 'MPTTRN'
C       SUBROUTINE FOR SCHEDULING TRAINER RIDES
C
        DO 1 I=1,NUMTRN
           N=I
           CALL TRNING
1       CONTINUE
        CALL MPTTRN
C
C       ALL SCHEDULING FUNCTIONS NOW COMPLETE, CALL
C       'STATS' SUBROUTINE TO GATHER MONTHLY STATISTICS
C
        CALL STATS
C
C       OUTPUT CREW SCHEDULING MATRIX
C
        WRITE(6,*)
        WRITE(6,*)'CREW SCHEDULING MATRIX'
        WRITE(6,*)
        WRITE(6,100)
100     FORMAT(7X,'1',3X,'2',3X,'3',3X,'4',3X,'5',3X,'6',3X,'7',3X,'8',3X,
       1'9',2X,'10',2X,'11',2X,'12,X,13',2X,'14',2X,'15',2X,'16',2X,'17
       2',2X,'18',2X,'19',2X,'20',2X,'21',2X,'22',2X,'23',2X,'24',2X,'25',
       32X,'26',2X,'27',2X,'28',2X,'29',2X,'30',2X,'31')
```

```
          FORMAT(1)='(29(I4))'
          FORMAT(2)='(30(I4))'
          FORMAT(3)='(31(I4))'
          FORMAT(4)='(32(I4))'
          FMT1=FORMAT(NODAYS-27)
          WRITE(6,FMT1)(I,(ISCHED(I,J),J=1,NODAYS),I=1,NOCRWS)
          WRITE(6,X)
C
C         OUTPUT MONTHLY DATA FOR EACH CREW
C
          WRITE(6,X)' MONTHLY STATISTICAL DATA: '
          WRITE(6,X)
          DO 5 I=1,NOCRWS
              WRITE(6,200)I,NOALTS(I),INTALT(I),NINT(FREDAY(I)),LVDAYS(I)
5         CONTINUE
          WRITE(6,X)
C
C         OUTPUT MEAN AND STANDARD DEVIATION FOR LINE
C         CREW DATA TOTALS
C
          WRITE(6,X)' NUMBER OF ALERTS:'
          WRITE(6,300)ALTMN,ALTSTD
          WRITE(6,X)' NUMBER OF INTEGRAL ALERTS:'
          WRITE(6,300)ALTIMN,ALTIST
          WRITE(6,X)' NUMBER OF FREE DAYS:'
          WRITE(6,300)FREMN,FRESTD
200       FORMAT(' CREW',I3,', ALERTS',I2,', INTEGRAL ALERTS',I2,', FREE DAY
         1S',I3,', ACTUAL NUMBER OF LEAVE DAYS ',I3)
300       FORMAT(' MEAN =',F7.3,5X,'STANDARD DEVIATION =',F8.5,/)
          RETURN
          END
C
C         XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C         X   SUBROUTINE 'TRNING' ASSIGNS CREWS TO X
C         X   CLASSROOM TRAINING REQUIREMENTS.  UP X
C         X   TO THREE TYPES OF CLASSROOM TRAINING X
C         X   MAY BE SCHEDULED.  THE SUBROUTINE    X
C         X   SCHEDULES UP TO 80% OF THE MAXIMUM   X
C         X   CLASS SIZE ON THE FIRST PASS THROUGH X
C         X   THE CREW FORCE, A SECOND PASS IS THENX
C         X   MADE STARTING FROM THE LAST AVAIL-   X
C         X   ABLE TRAINING DAY TO FILL UP TO THE  X
C         X   MAXIMUM CLASS SIZE.  NO CLASSES ARE  X
C         X   SCHEDULED FOR LESS THAN THE MINIMUM  X
C         X   CLASS SIZE AND A WARNING IS PRINTED  X
C         X   FOR ANY CREWS NOT SCHEDULED FOR EACH X
C         X   TYPE OF TRAINING                     X
C         XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C
          SUBROUTINE TRNING
          COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
         1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SL10),TNEXT,TNOW,XX(100)
          COMMON/UCOM1/IDAY,NXTDAY,IFILE,ISCHED(100,32),DUMMY(14),LVBEG(3),L
         1VEND(3),INTFLT(100),NOCRWS,NODAYS,NUMTRN,MAXLIN,MAXFCC,MAXSHP
```

```
          COMMON/UCOM2/ITRNDY(3,15),ITYPE(3),MRKDAY(3),MPTDAY(31),NUMMPT,MTB
        1MPT,N,MAXCLS(3),MINCLS(3)
          LOGICAL NOSCHD(100)
          DIMENSION ICREW(25),ICOUNT(15)
C
C       INITIALIZE SUBROUTINE PECULIAR ARRAYS
C
          DO 1 I=1,15
             ICOUNT(I)=0
1         CONTINUE
          DO 2 I=1,100
             NOSCHD(I)=.TRUE.
2         CONTINUE
          DO 3 I=1,25
             ICREW(I)=0
3         CONTINUE
C
C       BEGIN SEARCH OF THE CREW FORCE FOR EACH
C       AVAILABLE TRAINING DAY BEGINNING WITH THE
C       FIRST AVAILABLE DAY.  ('ITDAY' IS THE TRAINING DAY BEING CHECKED)
C
          DO 5 J=1,15
             ITDAY=ITRNDY(ITYPE(N),J)
             IF (ITDAY.EQ.0) GO TO 50
             DO 10 I=1,NOCRWS
C
C       LOGICAL ARRAY 'NOSCHED' IS USED TO RECORD
C       WHETHER CREW 'I' HAS BEEN SCHEDULED FOR THIS
C       TRAINING, .TRUE. INDICATES CREW HAS NOT BEEN
C       SCHEDULED.  IF CREW HAS NOT BEEN SCHEDULED
C       FOR THIS TRAINING AND IS NOT SCHEDULED FOR
C       ANY OTHER ACTIVITY, THEN INCREASE COUNT AND
C       RECORD CREW NUMBER IN 'ICREW' FOR THIS DAY.
C
                IF (NOSCHD(I).AND.ISCHED(I,ITDAY).EQ.0) THEN
                ICOUNT(J)=ICOUNT(J)+1
                ICREW(ICOUNT(J))=I
                ENDIF
C
C       AS SOON AS THE COUNT REACHES 80% OF THE
C       MAXIMUM CLASS SIZE, SCHEDULE THOSE CREWS BY
C       UPDATING THE SCHEDULING ARRAY AND CHANGE
C       'NOSCHED' TO .FALSE.  PROCEED TO NEXT AVAIL-
C       ABLE TRAINING DAY
C
                IF (ICOUNT(J).EQ.NINT(.8XMAXCLS(N))) THEN
                DO 15 K=1,ICOUNT(J)
                   ISCHED(ICREW(K),ITDAY)=MRKDAY(N)
                   NOSCHD(ICREW(K))=.FALSE.
15              CONTINUE
                GO TO 5
                ENDIF
10           CONTINUE
C
```

```
C       IF 80% OF MAX SIZE WAS NOT REACHED BUT MINIMUM
C       CLASS SIZE WAS REACHED, THEN SCHEDULE THOSE
C       CREWS
C
        IF (ICOUNT(J).GE.MINCLS(N)) THEN
           DO 20 K=1,ICOUNT(J)
              ISCHED(ICREW(K),ITDAY)=MRKDAY(N)
              NOSCHD(ICREW(K))=.FALSE.
2       CONTINUE
        ENDIF
5    CONTINUE
C
C    BEGIN SECOND SEARCH OF CREW FORCE STARTING WITH
C    LAST AVAILABLE DAY AND SCHEDULE UP TO THE
C    MAXIMUM CLASS SIZE
C
50   DO 25 I=1,NOCRWS
        IF (NOSCHD(I)) THEN
           DO 30 J=15,1,-1
           ITDAY=ITRNDY(ITYPE(N),J)
           IF (ITDAY.EQ.0) GO TO 30
           IF (ICOUNT(J).LT.MINCLS(N)) GO TO 30
           IF (ICOUNT(J).LT.MAXCLS(N) .AND. ISCHED(I,ITDAY).EQ.0) THEN
              ICOUNT(J)=ICOUNT(J)+1
              ISCHED(I,ITDAY)=MRKDAY(N)
              NOSCHD(I)=.FALSE.
              GO TO 25
           ENDIF
30      CONTINUE
        ENDIF
25   CONTINUE
C
C    PRINT ANY CREW NOT SCHEDULED FOR TRAINING
C
     DO 35 I=1,NOCRWS
        IF (NOSCHD(I)) THEN
           WRITE(6,100)I,MRKDAY(N)
        ENDIF
35   CONTINUE
C
C    PRINT RECAP OF TRAINING DAYS USED AND THE
C    NUMBER OF CREWS SCHEDULED FOR EACH DAY
C
     DO 40 I=1,15
        IF (ICOUNT(I).GT.(MINCLS(N)-1)) THEN
           WRITE(6,200)MRKDAY(N),ITRNDY(ITYPE(N),I),ICOUNT(I)
        ENDIF
40   CONTINUE
100  FORMAT(' CREW',I3,' NOT SCHEDULED FOR',I3,' TRAINING')
200  FORMAT(' FOR',I3,' TRAINING ON DAY',I3,',',I3,' CREWS SCHEDULED')
     RETURN
     END
```

```
C
C     ********************************************
C     *   SUBROUTINE 'MPTTRN' IS USED TO         *
C     *   SCHEDULE CREWS FOR UP TO TWO MPT        *
C     *   PERIODS PER MONTH.  WHEN SCHEDULING     *
C     *   MORE THAN ONE MPT, A USER SPECIFIED     *
C     *   MINIMUM TIME BETWEEN MPTS ('MTBMPT')    *
C     *   IS USED FOR TRAINING SEPARATION         *
C     ********************************************
C
      SUBROUTINE MPTTRN
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/UCOM1/IDAY,NXTDAY,IFILE,ISCHED(100,32),DUMMY(14),LVBEG(3),L
     1VEND(3),INTFLT(100),NOCRWS,NODAYS,NUMTRN,MAXLIN,MAXFCC,MAXSHP
      COMMON/UCOM2/ITRNDY(3,15),ITYPE(3),MRKDAY(3),MPTDAY(31),NUMMPT,MTB
     1MPT,N,MAXCLS(3),MINCLS(3)
      DIMENSION MPTSCH(100,2),NOASGN(31)
C
C     INITIALIZE SUBROUTINE PECULIAR ARRAYS
C
      DO 1 I=1,100
         DO 2 J=1,2
         MPTSCH(I,J)=0
2        CONTINUE
1     CONTINUE
      DO 3 I=1,31
         NOASGN(I)=0
3     CONTINUE
C
C     BEGIN SEARCH OF CREW FORCE FOR THE NUMBER OF
C     MPTS SPECIFIED.  SEARCH BEGINS WITH THE
C     LARGEST CREW NUMBER AND WORKS BACKWARDS
C     EACH CREW IS CHECKED AGAINST ALL DAYS BEFORE
C     MOVING TO THE NEXT CREW
C
      DO 5 K=1,NUMMPT
         DO 10 I=NOCRWS,1,-1
            DO 15 J=1,NODAYS
C
C     COMPARE THE NUMBER OF AVAILABLE MPT PERIODS FOR
C     DAY 'J' TO THE NUMBER OF CREWS ASSIGNED ON THAT DAY
C
            IF (MPTDAY(J).EQ.NOASGN(J)) GO TO 15
C
C     IF CREW IS AVAILABLE THEN SCHEDULE FOR MPT
C     IF THIS IS THE SECOND MPT THEN INSURE MINIMUM
C     TIME BETWEEN MPTS IS SATISFIED PRIOR TO
C     SCHEDULING CREW
C
            IF (ISCHED(I,J).EQ.0 .AND. MPTSCH(I,K).EQ.0) THEN
               IF (K.EQ.2.AND.ABS(J-MPTSCH(I,1)).LT.MTBMPT) GOTO 15
               ISCHED(I,J)=66
               MPTSCH(I,K)=J
```

A-54

```
                    NOASGN(J)=NOASGN(J)+1
                    GO TO 10
                 ENDIF
15             CONTINUE
C
C       PRINT CREWS NOT SCHEDULED FOR EACH MPT SESSION
C
         WRITE(6,X)' CREW ',I,' NOT SCHEDULED FOR MPT ',K
10          CONTINUE
5        CONTINUE
         RETURN
         END
C
C        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C        X   SUBROUTINE 'STATS' COLLECTS INDIVID- X
C        X   UAL CREW STATISTICS AND CUMULATIVE   X
C        X   STATISTICS ON LINE CREWS.  DATA IS   X
C        X   TAKEN FROM THE CREW SCHEDULING ARRAY X
C        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C
         SUBROUTINE STATS
         COMMON/UCOM1/IDAY,NXTDAY,IFILE,ISCHED(100,32),DUMMY(14),LVBEG(3),L
        1VEND(3),INTFLT(100),NOCRWS,NODAYS,NUMTRN,MAXLIN,MAXFCC,MAXSHP
         COMMON/UCOM3/NOALTS(100),FREDAY(100),INTALT(100),LVDAYS(100)
         COMMON/UCOM4/ALTMN,ALTSTD,ALTIMN,ALTIST,FREMN,FRESTD
C
C       INITIALIZE SUBROUTINE PECULIAR ARRAYS AND VARIABLES
C
         DATA NOALTS/100X0/
         DATA FREDAY/100X0/
         DATA INTALT/100X0/
         ALTMN=0.0
         ALTSTD=0.0
         ALTIMN=0.0
         ALTIST=0.0
         FREMN=0.0
         FRESTD=0.0
         CREWS=0.0
C
C       COLLECT DATA ON NUMBER OF ALERTS, NUMBER OF
C       INTEGRAL ALERTS AND NUMBER OF FREE DAYS FOR
C       EACH CREW. (NOTE: ACTUAL LEAVE DAYS ARE
C       ADDED TO FREE DAYS)
C
         DO 5 I=1,NOCRWS
            DO 10 J=1,NODAYS
            IF (ISCHED(I,J).GT.0 .AND. ISCHED(I,J).LE.15) THEN
               NOALTS(I)=NOALTS(I)+1
               IF (ABS(INTFLT(I)).EQ.ISCHED(I,J)) THEN
                  INTALT(I)=INTALT(I)+1
               ENDIF
            ELSE IF (ISCHED(I,J).EQ.0) THEN
               FREDAY(I)=FREDAY(I)+1
            ENDIF
```

A-55

```
10        CONTINUE
          FREDAY(I)=FREDAY(I)+LVDAYS(I)
5         CONTINUE
C
C         COLLECT CUMULATIVE STATISTICS FOR LINE CREWS
C         ONLY.  DATA CONTAINED IN ARRAY 'INTFLT' IS
C         USED TO SEPARATE LINE CREWS FROM FLIGHT
C         COMMANDERS, CODED AS NEGATIVE NUMBERS, AND
C         DOV/DOT CREWS, CODED AS 16 AND 17
C
          DO 15 I=1,NOCRWS
             IF (INTFLT(I).GT.0 .AND. INTFLT(I).LE.15) THEN
                ALTMN=ALTMN+REAL(NOALTS(I))
                ALTSTD=ALTSTD+REAL(NOALTS(I)**2)
                ALTIM'=ALTIMN+REAL(INTALT(I))
                ALTIST=ALTIST+REAL(INTALT(I)**2)
                FREMN=FREMN+FREDAY(I)
                FRESTD=FRESTD+FREDAY(I)**2
                CREWS=CREWS+1
             ENDIF
15        CONTINUE
C
C         COMPUTE MEAN AND STANDARD DEVIATION FOR
C         CUMULATIVE STATISTICS
C
          ALTMN=ALTMN/CREWS
          ALTSTD=(ALTSTD-CREWS*ALTMN**2)/(CREWS-1)
          IF (ALTSTD.GT.0.0) ALTSTD=SQRT(ALTSTD)
          ALTIMN=ALTIMN/CREWS
          ALTIST=(ALTIST-CREWS*ALTIMN**2)/(CREWS-1)
          IF (ALTIST.GT.0.0) ALTIST=SQRT(ALTIST)
          FREMN=FREMN/CREWS
          FRESTD=(FRESTD-CREWS*FREMN**2)/(CREWS-1)
          IF (FRESTD.GT.0.0) FRESTD=SQRT(FRESTD)
          RETURN
          END
```

LIST OF VARIABLES: SLAM SCHEDULING PROGRAM

ARRAYS

DUMMY(14) - zero array; real

FORMAT(4) - stores format parameters for output of scheduling array;
character

FREDAY(100) - number of free days for each crew; real

ICOUNT(15) - number of crews selected for classroom type training on a
given day; integer

A-56

ICREW(25) – crew numbers of crews assigned to classroom training on a particular day; integer

INTALT(100) – number of integral flight alerts for each crew; integer

INTFLT(100) – stores coded flight assignment data for each crew; integer

ISCHED(100,32) – crew scheduling matrix; integer

ITRNDY(3,15) – available training days for each type of classroom training; integer

ITYPE(3) – identifies type of classroom training; integer

LVBEG(3) – begin leave dates for each crew, attributes 7, 9, 11; integer

LVEND(3) – end leave dates for each crew, attributes 8, 10, 12; integer

LVDAYS(100) – number of actual leave days per crew; integer

MAXCLS(3) – maximum class size for each type classroom training; integer

MINCLS(3) – minimum class size for each type classroom training; integer

MPTDAY(31) – number of available MPT sessions per day; integer

MPTSCH(100,2) – date of assignment for each MPT session by crew number; integer

MRKDAY(3) – numeric code of each type classroom training; integer

NOALTS(100) – number of alerts for each crew; integer

NOASGN(31) – number of crews actually assigned to MPT per day; integer

NOSCHD(100) – crews not scheduled for classroom training; logical

VARIABLES

ALTIMN – mean number of integral alerts; real

ALTIST – integral alert standard deviation; real

ALTMN – mean number of alerts; real

ALTSTD – alert standard deviation; real

CREWS – number of line crews; real

DURLV – duration of leave period; real

FCCPCT - percentage of flight commanders sent to non-SCP alerts; real

FLTPCT - percentage of crews sent to their assigned flight for non-SCP alerts; real

FMT1 - output format; character

FREMN - mean number of free days; real

FRESTD - free days standard deviation; real

FTPCTS - percentage of crews sent to their assigned flight for SCP alerts; real

IDAY - integer representation of SLAM variable TNOW; integer

IFILE - alert queue identifier; integer

ISTRM1 - random number stream used for non-SCP alerts; integer

ISTRM2 - random number stream used for SCP alerts; integer

ITDAY - classroom training day under consideration; integer

MAXFCC - maximum number of alerts for flight commander; integer

MAXLIN - maximum number of alerts for line crews; integer

MAXSHP - maximum number of alerts for DOV/DOT crews; integer

MOVE - temporary pointer holder in subroutine LEAVE; integer

MTBMPT - minimum time between MPT training; integer

N - identifies type classroom training; integer

NEXT - pointer used when searching EA queue; integer

NOCRWS - total number of crews; integer

NODAYS - number of days in month; integer

NUMMPT - number of MPT sessions requested per crew; integer

NUMTRN - number of classroom type training; integer

NXTDAY - IDAY + 1; integer

RN - random number drawn by SLAM function DRAND; real

SHPCTS - percentage of DOV/DOT crews sent to SCP alerts; real

SHPPCT - percentage of DOV/DOT crews sent to non-SCP alerts; real
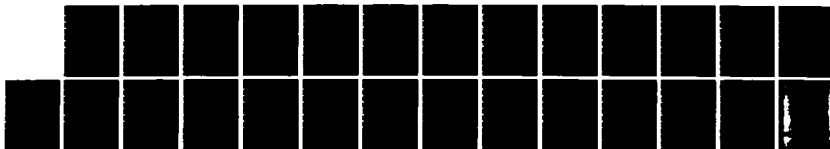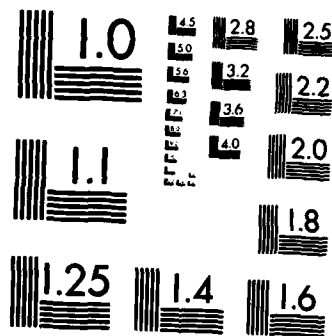
MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

APPENDIX B

This appendix contains the data and calculations used in
the experiments and sensitivity analysis presented in Chap-
ter IV. Note that the alpha level for all statistical tests
used in this appendix is 0.05.

Table B-1. Data Points for Factors One, Two, and Three
versus Permformance Measure One

Factor 1	PM1	Factor 2	PM1	Factor 3	PM3
0.0	0.899	0.0	0.934	0.0	0.910
0.1	0.940	0.1	0.954	0.1	0.918
0.2	0.943	0.2	0.951	0.2	0.924
0.3	0.938	0.3	0.939	0.3	0.935
0.4	0.929	0.4	0.933	0.4	0.941
0.5	0.930	0.5	0.926	0.5	0.943
		0.6	0.925	0.6	0.945
		0.7	0.916	0.7	0.958
		0.8	0.915	0.8	0.938
		0.9	0.910	0.9	0.932
		1.0	0.910	1.0	0.890

Table B-2. Data Points for Factor Three and
           Factor One versus Performance Measures
           One, Two and Three

Factor Three	Factor One	PM1	PM2	PM3
0.0	0.0	0.910	0.826	0.775
0.0	0.1	0.910	0.825	0.777
0.0	0.2	0.920	0.830	0.780
0.0	0.3	0.910	0.835	0.793
0.0	0.4	0.910	0.838	0.795
0.0	0.5	0.913	0.838	0.788
0.1	0.1	0.915	0.841	0.803
0.1	0.3	0.918	0.866	0.836
0.1	0.5	0.918	0.858	0.818
0.2	0.0	0.910	0.858	0.820
0.2	0.2	0.928	0.860	0.829
0.2	0.3	0.923	0.858	0.829
0.2	0.4	0.924	0.865	0.833
0.2	0.5	0.930	0.868	0.838
0.3	0.1	0.930	0.878	0.848
0.3	0.3	0.940	0.875	0.850
0.4	0.1	0.936	0.884	0.862
0.4	0.2	0.941	0.892	0.878
0.4	0.3	0.942	0.891	0.875
0.4	0.4	0.943	0.900	0.883
0.4	0.5	0.940	0.886	0.858
0.5	0.1	0.944	0.900	0.886
0.5	0.2	0.944	0.898	0.878
0.5	0.3	0.944	0.898	0.882
0.5	0.4	0.942	0.892	0.878
0.5	0.5	0.940	0.888	0.868
0.6	0.1	0.955	0.919	0.908
0.6	0.2	0.958	0.929	0.928
0.6	0.3	0.959	0.929	0.925
0.7	0.1	0.958	0.932	0.930
0.7	0.2	0.950	0.932	0.930
0.7	0.3	0.958	0.931	0.928
0.8	0.0	0.871	0.860	0.853
0.8	0.1	0.956	0.948	0.956
0.8	0.2	0.961	0.951	0.955
0.9	0.0	0.932	0.924	0.924
0.9	0.1	0.934	0.924	0.926
1.0	0.0	0.890	0.879	0.874

Table B-3. Data Points for Factor Three and
Factor Two versus Performance Measures
One, Two and Three

Factor Three	Factor Two	PM1	PM2	PM3
0.0	0.5	0.913	0.838	0.788
0.0	0.6	0.910	0.838	0.795
0.0	0.7	0.917	0.838	0.798
0.0	0.8	0.910	0.825	0.773
0.0	0.9	0.910	0.825	0.777
0.0	1.0	0.910	0.828	0.778
0.1	0.3	0.920	0.848	0.813
0.1	0.5	0.920	0.850	0.812
0.1	0.6	0.916	0.838	0.800
0.1	0.8	0.915	0.848	0.808
0.2	0.3	0.915	0.868	0.838
0.2	0.4	0.930	0.858	0.825
0.2	0.5	0.925	0.858	0.828
0.2	0.6	0.930	0.860	0.830
0.2	0.8	0.910	0.858	0.820
0.3	0.3	0.940	0.875	0.850
0.3	0.6	0.930	0.878	0.848
0.4	0.1	0.940	0.886	0.858
0.4	0.2	0.941	0.893	0.878
0.4	0.3	0.942	0.891	0.875
0.4	0.4	0.941	0.895	0.880
0.4	0.5	0.936	0.884	0.862
0.5	0.0	0.938	0.888	0.868
0.5	0.1	0.942	0.892	0.878
0.5	0.2	0.944	0.898	0.882
0.5	0.3	0.944	0.898	0.878
0.5	0.4	0.944	0.900	0.886
0.6	0.1	0.957	0.922	0.912
0.6	0.2	0.958	0.929	0.928
0.6	0.3	0.955	0.922	0.916
0.7	0.0	0.958	0.931	0.928
0.7	0.1	0.960	0.932	0.932
0.7	0.2	0.958	0.932	0.930
0.8	0.0	0.934	0.923	0.923
0.8	0.1	0.964	0.948	0.956
0.8	0.2	0.875	0.865	0.863
0.9	0.0	0.933	0.924	0.926
0.9	0.1	0.932	0.924	0.924
1.0	0.0	0.890	0.879	0.874

Table B-4. Best Data Points for Factors
            One, Two, and Three for Performance
            Measure One

Factor Level	Number of Observations	Data Points F1	F2	F3	PM1 Mean	Variance
1	10	0.1	0.1	0.6	0.954	4.87E-5
2	10	0.1	0.3	0.6	0.955	2.82E-5
3	10	0.1	0.2	0.7	0.958	4.09E-5
4	14	0.1	0.1	0.8	0.964	2.42E-5
5	10	0.1	0.0	0.9	0.933	4.75E-5
6	18	0.2	0.2	0.6	0.958	6.19E-5
7	12	0.2	0.1	0.7	0.960	7.30E-5
8	10	0.2	0.0	0.8	0.961	9.85E-5
9	10	0.3	0.1	0.6	0.959	9.54E-6
10	12	0.3	0.0	0.7	0.958	3.35E-5

ANALYSIS FOR FACTORS ONE, TWO, AND THREE

The ten factor levels were compared to check for significant differences in the means. First the variances of the factor levels were compared. If the variances could be considered equivalent by using an F-test, the means were checked using a T-test. If the variances could not be considered equivalent then a Wilcoxon-Mann-Whitney rank sum test was used to test for a difference in the means. The results are presented in Table B-5. Factor level four had the highest mean, therefore all other factor levels were compared to it. The mean for factor level four was 0.964 and the variance was 2.42E-5. Only factor level eight had to be tested with the rank sum test. The results of this test showed that there was no significant difference in the means of factor levels four and eight. These two factor levels (.1,.1,.8 and .2,0,.8) are the best levels for factors one, two, and three for performance measure one.

Table B-5. Analysis for Factors One, Two, and
Three for Performance Measure One

Factor Level	Mean	Variance	D.F.	Computed F-value	D.F.	Computed T-value
1	0.954	4.87E-5	9,13	2.01	22	-13.27
2	0.955	2.82E-5	9,13	1.17	22	-14.08
3	0.958	4.09E-5	9,13	1.69	22	-8.42
5	0.933	4.75E-5	9,13	1.96	22	-41.47
6	0.958	6.19E-5	13,17	2.56	30	-9.94
7	0.960	7.30E-5	11,13	3.02	24	-5.05
8	0.961	9.85E-5	XXUsed rank sum testXX			
9	0.959	9.54E-6	9,13	2.54	22	-9.75
10	0.958	3.35E-5	11,13	1.38	24	-9.84

Table B-6.   Best Data Points for Factors
             One, Two, and Three for Performance
             Measure Two

Factor Level	Number of Observations	Data Points F1	F2	F3	PM1 Mean	Variance
1	10	0.1	0.1	0.6	0.915	9.39E-5
2	10	0.1	0.3	0.6	0.922	8.43E-5
3	10	0.1	0.2	0.7	0.932	4.83E-5
4	14	0.1	0.1	0.8	0.948	1.79E-5
5	10	0.1	0.0	0.9	0.924	1.16E-4
6	18	0.2	0.2	0.6	0.929	1.74E-5
7	12	0.2	0.1	0.7	0.932	1.76E-4
8	10	0.2	0.0	0.8	0.951	5.47E-5
9	10	0.3	0.1	0.6	0.929	9.54E-6
10	12	0.3	0.0	0.7	0.931	8.10E-5

In the analysis for this performance measure, factor level eight has the highest mean (0.951) with a variance of 5.47E-5. All other factor levels were compared to level eight. Factor levels four, six, and nine had to be tested with the rank sum test. The results are summarized in Tables B-7 and B-8. Once again factor levels four and eight were found to be the best levels for factor one, two, and three.

Table B-7.   Analysis for Factors One, Two, and
             Three for Performance Measure Two

Factor Level	Mean	Variance	D.F.	Computed F-value	D.F.	Computed T-value
1	0.915	9.39E-5	9,9	1.72	18	-28.02
2	0.922	8.43E-5	9,9	1.54	18	-23.33
3	0.932	4.83E-5	9,9	1.13	18	-17.76
4	0.948	1.79E-5	**Used rank sum test**			
5	0.924	1.16E-4	9,9	2.12	18	-19.61
6	0.929	1.74E-5	**Used rank sum test			
7	0.932	1.76E-4	9,11	3.22	20	-13.07
9	0.929	9.54E-6	**Used rank sum test**			
10	0.931	8.10E-5	9,11	1.48	20	-17.93

Table B-8.   Rank Sum Test Results for Factors
             One, Two, and Three and Performance
             Measure Two

Factor Level	Expected Value	Variance	Sum of Ranks	D.F.	Computed Z-value
4	125	291.67	136.5	22	-0.673
6	261	435	171	30	-4.32
9	105	175	55	22	-3.78

Table B-9.   Best Data Points for Factors
             One, Two, and Three for Performance
             Measure Three

Factor Level	Number of Observations	Data Points F1	F2	F3	PM1 Mean	Variance
1	10	0.1	0.1	0.6	0.899	1.34E-3
2	10	0.1	0.3	0.6	0.916	2.05E-4
3	10	0.1	0.2	0.7	0.930	1.78E-4
4	14	0.1	0.1	0.8	0.956	2.38E-5
5	10	0.1	0.0	0.9	0.926	2.04E-4
6	18	0.2	0.2	0.6	0.928	6.93E-5
7	12	0.2	0.1	0.7	0.932	2.51E-4
8	10	0.2	0.0	0.8	0.955	5.00E-5
9	10	0.3	0.1	0.6	0.925	5.07E-5
10	12	0.3	0.0	0.7	0.928	1.79E-4

Factor level four has the highest mean for performance measure three (0.956) with a variance of 2.38E-5. Only two factor levels can be compared without using a rank sum test, factor levels 8 and 9. The results of the analysis are presented in Tables B-10 and B-11. The results show, once again, that factor levels four and eight are the best levels for performance measrue three. These results indicate that the same factor levels can be used for all three performance measures.

Table B-10. Analysis for Factors One, Two, and Three for Performance Measure Three

Factor Level	Mean	Variance	D.F.	Computed F-value	D.F.	Computed T-value
8	0.955	5.00E-5	9,13	2.10	22	-1.32
9	0.929	9.54E-6	9,13	2.13	22	-52.97

Table B-11. Rank Sum Test Results for Factors One, Two, and Three and Performance Measure Three

Factor Level	Expected Value	Variance	Sum of Ranks	D.F.	Computed Z-value
1	125	291.67	55.0	22	-4.10
2	125	291.67	55.0	22	-4.10
3	125	291.67	60.0	22	-3.81
5	125	291.67	60.0	22	-3.81
6	297	693.00	155.5	30	-5.38
7	162	378.00	64.5	24	-5.01
10	162	378.00	78.0	24	-4.32

Table B-12. Factor Levels for Factors One, Two, and
Three Used in Testing Factors Four and
Five

Factor Level	Factor 1	Data Points Factor 2	Factor 3
1	0.1	0.2	0.7
2	0.2	0.1	0.7
3	0.3	0.0	0.7
4	0.1	0.4	0.5
5	0.4	0.1	0.5
6	0.2	0.3	0.5
7	0.3	0.2	0.5
8	0.5	0.0	0.5
9	0.1	0.5	0.4
10	0.5	0.1	0.4

Table B-13. Factor Levels for Factors Four and Five
Used in Finding the Best Factor Four and
Five Levels

Factor Level	Data Points Factor 4	Factor 5
1	0.1	0.9
2	0.2	0.8
3	0.3	0.7
4	0.4	0.6
5	0.6	0.4
6	0.8	0.2
7	1.0	0.0

Table B-14. Results From Data Runs for Factors Four
and Five for Performance Measure One

Factors 1,2,and 3 Levels	Factors 4 and 5 Levels						
	1	2	3	4	5	6	7
1	0.96	0.97	0.96	0.96	0.96	0.95	0.95
2	0.96	0.96	0.96	0.95	0.95	0.95	0.95
3	0.96	0.96	0.96	0.94	0.95	0.95	0.96
4	0.95	0.97	0.96	0.94	0.94	0.94	0.94
5	0.95	0.95	0.95	0.94	0.94	0.94	0.94
6	0.95	0.95	0.95	0.94	0.95	0.94	0.94
7	0.95	0.95	0.95	0.94	0.94	0.95	0.95
8	0.95	0.95	0.94	0.94	0.94	0.94	0.93
9	0.95	0.95	0.94	0.94	0.93	0.93	0.93
10	0.95	0.95	0.94	0.94	0.94	0.93	0.94
Mean	.953	.956	.951	.943	.944	.942	.943
Variance (xE-5)	2.30	7.16	7.70	4.50	7.18	6.30	9.00

The analysis for determining the best factor levels of
factors four and five was conducted in the same manner as
the analysis for factors one, two, and three. Table B-15
summarizes the results of the analysis for performance
measure one. Factor level two has the highest mean (0.956)
and a variance of 7.16E-5. The degrees of freedom for all
of the tests performed in this section are (9,9) and 16 for
the F-test and the T-test, respectively. The results show
that this factor level is better than all other factor
levels tested. Setting factor four at 0.2 and factor five
at 0.8 achieves the best results for performance measure
one.

Table B-15. Results for Factors Four and Five
for Performance Measure One

Factor Level	Mean	Variance	Computed F-value	Computed T-value
1	0.953	2.30E-5	3.11	-2.93
3	0.951	7.70E-5	1.08	-13.88
4	0.943	4.50E-5	1.59	-11.42
5	0.944	7.18E-5	1.00	-11.42
6	0.942	6.30E-5	1.14	-9.51
7	0.943	9.00E-5	1.20	-10.45

Table B-16. Results From Data Runs for Factors Four
and Five for Performance Measure Two

Factors 1,2,and 3 Levels	Factors 4 and 5 Levels						
	1	2	3	4	5	6	7
1	0.94	0.94	0.94	0.93	0.93	0.93	0.93
2	0.94	0.94	0.93	0.92	0.92	0.92	0.91
3	0.93	0.93	0.93	0.92	0.92	0.92	0.92
4	0.92	0.94	0.94	0.90	0.90	0.89	0.89
5	0.90	0.90	0.90	0.90	0.89	0.89	0.88
6	0.91	0.91	0.91	0.90	0.90	0.89	0.89
7	0.91	0.91	0.90	0.90	0.89	0.90	0.89
8	0.90	0.90	0.89	0.89	0.88	0.89	0.88
9	0.90	0.91	0.89	0.89	0.88	0.88	0.87
10	0.90	0.90	0.89	0.89	0.89	0.87	0.88
Mean	.915	.918	.912	.904	.899	.898	.894
Variance (xE-4)	2.70	3.10	4.40	2.04	3.11	3.70	3.82

Factor level two has the highest mean (0.918) with a variance of 3.1E-4 for performance measure two. Factor levels one and two were found to have statistically equivalent means. Both of these factor levels are the best factor settings for factors four and five for performance measure two. The results are summarized in Table B-17. Again, the degrees of freedom are (9,9) and 18 for the F-test and T-test respectively for all tests.

Table B-17.   Results for Factors Four and Five
for Performance Measure Two

Factor Level	Mean	Variance	Computed F-value	Computed T-value
1	0.915	2.70E-4	1.15	-1.18
3	0.912	4.40E-4	1.42	-2.08
4	0.904	2.04E-4	1.52	-5.86
5	0.899	3.11E-4	1.00	-7.23
6	0.898	3.70E-4	1.19	-7.28
7	0.894	3.82E-4	1.23	-8.66

Table B-18. Results From Data Runs for Factors Four
and Five for Performance Measure Three

Factors 1,2,and 3 Levels	Factors 4 and 5 Levels						
	1	2	3	4	5	6	7
1	0.95	0.95	0.94	0.93	0.93	0.92	0.91
2	0.94	0.94	0.93	0.92	0.91	0.91	0.91
3	0.94	0.94	0.92	0.92	0.91	0.91	0.91
4	0.91	0.95	0.94	0.89	0.89	0.87	0.87
5	0.89	0.89	0.89	0.89	0.88	0.87	0.86
6	0.90	0.90	0.90	0.89	0.89	0.87	0.87
7	0.90	0.90	0.89	0.88	0.88	0.88	0.87
8	0.89	0.89	0.88	0.87	0.86	0.87	0.85
9	0.89	0.89	0.88	0.87	0.86	0.85	0.84
10	0.88	0.88	0.87	0.86	0.86	0.84	0.85
Mean	.909	.913	.904	.892	.887	.879	.874
Variance (xE-4)	6.32	8.00	6.90	5.70	5.80	7.00	7.20

Factor level two still has the highest mean (0.913) with a variance of 8.00E-4. A summary of the results is presented in Table B-19. Factor levels two and four were found to be the best factor settings. This shows that factor level 2 (factor four set to 0.2 and factor five set to 0.8) is the best factor setting for all three performance measures.

Table B-19.   Results for Factors Four and Five
for Performance Measure Three

Factor Level	Mean	Variance	Computed F-value	Computed T-value
1	0.909	6.32E-4	1.27	-1.00
3	0.904	6.90E-4	1.16	-2.21
4	0.892	5.70E-4	1.40	-5.38
5	0.887	5.80E-4	1.38	-6.64
6	0.879	7.00E-4	1.14	-8.33
7	0.874	7.20E-4	1.11	-9.49

Table 8-20.   Performance Measure Values for Maximum
versus Minimum SCP Force for Performance
Measure One

Data Points F1	F2	F3	Maximum SCP PM Value	Minimum SCP PM Value
0.0	0.8	0.0	0.908	0.910
0.0	0.8	0.2	0.920	0.910
0.1	0.6	0.1	0.913	0.915
0.1	0.6	0.3	0.930	0.930
0.1	0.8	0.1	0.910	0.915
0.2	0.4	0.2	0.920	0.925
0.2	0.4	0.4	0.938	0.940
0.2	0.6	0.2	0.925	0.930
0.2	0.8	0.0	0.908	0.910
0.3	0.3	0.1	0.918	0.918
0.3	0.3	0.3	0.930	0.930
0.3	0.5	0.1	0.918	0.920
0.3	0.5	0.2	0.923	0.925
0.3	0.6	0.1	0.910	0.948
0.3	0.7	0.0	0.910	0.910
0.4	0.4	0.2	0.920	0.926
0.4	0.6	0.0	0.910	0.910
0.5	0.3	0.1	0.918	0.923
0.5	0.3	0.2	0.920	0.930
0.5	0.5	0.0	0.918	0.913
0.0	1.0	0.0	0.908	0.910

Table B-21. Performance Measure Values for Maximum
versus Minimum SCP Force for Performance
Measure Two

| Data Points | | | Maximum SCP | Minimum SCP |
F1	F2	F3	PM Value	PM Value
0.0	0.8	0.0	0.820	0.825
0.0	0.8	0.2	0.850	0.858
0.1	0.6	0.1	0.838	0.840
0.1	0.6	0.3	0.865	0.878
0.1	0.8	0.1	0.832	0.848
0.2	0.4	0.2	0.853	0.860
0.2	0.4	0.4	0.885	0.895
0.2	0.6	0.2	0.855	0.860
0.2	0.8	0.0	0.813	0.830
0.3	0.3	0.1	0.838	0.845
0.3	0.3	0.3	0.868	0.875
0.3	0.5	0.1	0.840	0.850
0.3	0.5	0.2	0.855	0.858
0.3	0.6	0.1	0.833	0.912
0.3	0.7	0.0	0.830	0.835
0.4	0.4	0.2	0.853	0.865
0.4	0.6	0.0	0.828	0.838
0.5	0.3	0.1	0.840	0.858
0.5	0.3	0.2	0.852	0.868
0.5	0.5	0.0	0.822	0.838
0.0	1.0	0.0	0.825	0.828

Table B-22.   Performance Measure Values for Maximum
versus Minimum SCP Force for Performance
Measure Three

| Data Points | | | Maximum SCP | Minimum SCP |
F1	F2	F3	PM Value	PM Value
0.0	0.8	0.0	0.770	0.773
0.0	0.8	0.2	0.813	0.820
0.1	0.6	0.1	0.715	0.800
0.1	0.6	0.3	0.835	0.848
0.1	0.8	0.1	0.793	0.808
0.2	0.4	0.2	0.815	0.828
0.2	0.4	0.4	0.865	0.880
0.2	0.6	0.2	0.825	0.830
0.2	0.8	0.0	0.765	0.780
0.3	0.3	0.1	0.800	0.808
0.3	0.3	0.3	0.838	0.850
0.3	0.5	0.1	0.800	0.813
0.3	0.5	0.2	0.825	0.828
0.3	0.6	0.1	0.793	0.888
0.3	0.7	0.0	0.778	0.793
0.4	0.4	0.2	0.818	0.833
0.4	0.6	0.0	0.788	0.795
0.5	0.3	0.1	0.798	0.818
0.5	0.3	0.2	0.818	0.858
0.5	0.5	0.0	0.772	0.788
0.0	1.0	0.0	0.773	0.778

The analysis of the data for a maximum SCP force was conducted in the same manner as the rest of the results. The results are summarized in Table B-23. The degrees of freedom used in the table are (20,20) and 40 for the F-test and T-test respectively. The results show that the minimum SCP force has a better mean value for all three performance measures.

Table B-23. Results of Analysis of Maximum versus Minimum SCP Force

Performance	Mean		Variance (xE-4)		F	T
Measure	Min	Max	Min	Max	Value	Value
1	.921	.918	1.17	.679	1.72	-4.52
2	.855	.843	4.87	3.19	1.53	-8.66
3	.820	.800	10.2	10.4	1.02	-9.07

Table B-24. Performance Measure Values for Leave versus Non-Leave Schedules for Performance Measure One

Data Points			Leave	No Leave
F1	F2	F3	PM Value	PM Value
0.2	0.4	0.2	0.893	0.925
0.2	0.4	0.4	0.908	0.940
0.2	0.6	0.2	0.898	0.930
0.3	0.3	0.1	0.885	0.918
0.3	0.3	0.3	0.905	0.930
0.3	0.5	0.1	0.885	0.920
0.4	0.4	0.2	0.898	0.926
0.4	0.6	0.0	0.885	0.910
0.5	0.3	0.1	0.888	0.923
0.5	0.3	0.2	0.903	0.930
0.5	0.5	0.0	0.863	0.913

Table B-25. Performance Measure Values for Leave versus
            Non-Leave Schedules for Performance
            Measure Two

| Data Points | | | Leave | No Leave |
F1	F2	F3	PM Value	PM Value
0.2	0.4	0.2	0.830	0.860
0.2	0.4	0.4	0.860	0.895
0.2	0.6	0.2	0.835	0.860
0.3	0.3	0.1	0.820	0.845
0.3	0.3	0.3	0.850	0.875
0.3	0.5	0.1	0.823	0.850
0.4	0.4	0.2	0.838	0.865
0.4	0.6	0.0	0.810	0.838
0.5	0.3	0.1	0.820	0.858
0.5	0.3	0.2	0.838	0.868
0.5	0.5	0.0	0.793	0.838

Table B-26. Performance Measure Values for Leave versus
            Non-Leave Schedules for Performance
            Measure Three

| Data Points | | | Leave | No Leave |
F1	F2	F3	PM Value	PM Value
0.2	0.4	0.2	0.790	0.828
0.2	0.4	0.4	0.828	0.880
0.2	0.6	0.2	0.790	0.830
0.3	0.3	0.1	0.773	0.808
0.3	0.3	0.3	0.813	0.850
0.3	0.5	0.1	0.770	0.813
0.4	0.4	0.2	0.803	0.833
0.4	0.6	0.0	0.758	0.795
0.5	0.3	0.1	0.770	0.818
0.5	0.3	0.2	0.793	0.858
0.5	0.5	0.0	0.745	0.788

The analysis for inserting an assumed leave distribution into the scheduling program is in the same format as the analysis of the maximum SCP force. The degrees of freedom for the analysis are (10,10) and 20 for the F-test and the T-test, respectively. The results, presented in Table B-27, show that the schedule with no leave distribution has a higher average performance measure value than a schedule with an assumed leave distribution for all three performance measures.

Table B-27.   Results of Analysis of Leave vs.Non-Leave
             Schedules

| Performance | Mean | | Variance (xE-4) | | F | T |
Measure	Lv	No Lv	Lv	No Lv	Value	Value
1	.892	.924	1.62	.734	2.21	-21.87
2	.829	.859	3.45	2.83	1.22	-20.93
3	.785	.827	5.99	7.53	1.26	-11.98

Vita

David Hollis Berg was born on 6 November 1954 in Oakland, California. He graduated from high school in Concord, California in 1972 and attended the United States Air Force Academy, Colorado Springs, Colorado from which he received the degree of Bachelor of Science in Behavioral Science in June 1976. Upon graduation he was commissioned in the United States Air Force and entered active duty. He completed Initial Qualification Training in missile operations in November of 1976 and was assigned to the 341st Strategic Missile Wing, Malmstrom AFB, Montana as a missile combat crew member. He served as a deputy commander, commander, and instructor while a crew member. In 1981 he was assigned as a missile operations instructor at the 4315th Combat Crew Training Squadron, Vandenberg AFB, California where he served until entering the School of Engineering, Air Force Institute of Technology, in August of 1982.

Permanent Address: 839 San Simeon Dr.

Concord, California

94518

Vita

Rodney George Nuss was born on 8 April 1952 in New
Orleans, Louisiana. He graduated from The Holy Cross High
School in New Orleans, Louisiana in 1970 and attended Tulane
University, New Orleans, Louisiana, from which he received
the degree of Bachelor of Science in Mathematics in May
1974. Upon graduation he received a commission in the
United States Air Force through the ROTC program. He en-
tered active duty in July 1974 as an accounting and finance
officer assigned to Craig AFB, Alabama. Following a remote
tour at Thule AB, Greenland as the base comptroller, he
entered the missile operations career field in December
1976. A distinguished graduate of missile combat crew
Initial Qualification Training, he was assigned to the 341st
Strategic Missile Wing, Malmstrom AFB, Montana in March
1977. His experience as a missile combat crew member
includes deputy commander, commander, evaluator, and SAC
Missile Combat Competition positions. Most recently, he
served as the chief of Instructional Systems Development at
the 341st Strategic Missile Wing until entering the School
of Engineering, Air Force Institute of Technology, in August
of 1982.

Permanent Address: P.O. Box 109

Magnolia, Mississippi

39652

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	Approved for public release; distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GST/OS/84M-2	5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION School of Engineering	6b. OFFICE SYMBOL (If applicable) AFIT/ENS	7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS (City, State and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB, Ohio 45433	7b. ADDRESS (City, State and ZIP Code)

8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS.			
	PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
11. TITLE (Include Security Classification) see box 19				

12. PERSONAL AUTHOR(S) David H. Berg, Capt, USAF	Rodney G. Nuss, Capt, USAF

13a. TYPE OF REPORT MS Thesis	13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (Yr., Mo., Day) 1984 March	15. PAGE COUNT 215

16. SUPPLEMENTARY NOTATION

~~Approved for public release IAW AFR 190-17.~~
~~LYNN E. WOLAVER~~
~~Dean for Research and Professional Development~~
~~Air Force Institute of Technology (ATC)~~
~~Wright-Patterson AFB OH 45433~~

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	Scheduling, Simulation, Personnel (Guided Missile)
12	01		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: An Improved Missile Combat Crew Scheduling System Using the Simulation Language for Alternative Methods (SLAM)

Thesis Chairman: Thomas D. Clark, Lt Col, USAF

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL Thomas D. Clark, Lt Col, USAF	22b. TELEPHONE NUMBER (Include Area Code) 513-255-3362	22c. OFFICE SYMBOL AFIT/ENS

DD FORM 1473, 83 APR   EDITION OF 1 JAN 73 IS OBSOLETE.   UNCLASSIFIED

A missile combat crew scheduling program was developed using the SLAM simulation language. The program schedules missile combat crew members for alerts, training, and leave. The program was designed to build monthly schedules for a three-squadron strategic missile wing.

A performance measure was developed to evaluate alternative schedules that were developed by the program. The measure was developed by using Multi-Criteria Decision Theory techniques and alternatives were compared using response surface methodology. The performance measure is general in nature and can be modified to apply to any decision maker.

Analysis was conducted to determine the best internal factor settings for the program for a given performance measure. These internal factor settings are used to determine the next crew selected for alert.

Sensitivity analyses were conducted that include evaluating two additional performance measures, adding an assumed leave distribution, and changing the structure of the crew force.

The program develops a feasible monthly schedule that meets the performance criteria. The internal factors were found to be robust across all three performance measures examined.

END

FILMED

DTIC